Tangible Spatial Augmented Reality in
Rapid Prototyping: Multiple and
Differential Tangible Object
Manipulation and Interaction

TIM SIMON

Degree project in
Communication Systems
Second level, 30.0 HEC
Stockholm, Sweden
Tangible Spatial Augmented Reality in Rapid Prototyping: Multiple and differential tangible object manipulation and interaction

Author: Tim SIMON

Supervisor: Dr. Ross SMITH
            Mark SMITH

October 22, 2012
0.1 Abstract

Tangible Interface Objects underpin the interactions between users and a SAR environment. When utilizing SAR for rapid-prototyping work flows, particularly when the subject of the prototyping is a user-input centric design, the role of the Tangible Interface Objects is crucial. A Tangible Interface Object with form or functionality that does not reflect that of its real-world counterpart is detrimental to the prototyping work flow, where realism in prototypes is highly sought after. Moving from the use of ‘dumb’ input controls with SAR-emulated functionality to ‘intelligent’, state-aware input controls can greatly aid the rapid-prototyping work flow, and SAR environments generally. This research examines two areas: integrating sensors into input controls to enhance both the self-awareness and the local environmental-awareness of the input control, and increasing state-awareness of traditional input controls such as switches and radial dials. This second area has a focus on input controls which do not require a traditional power source. The results from both these areas demonstrate that ‘intelligent’ Tangible Interface Objects are viable, providing numerous benefits to SAR scenes, particularly in the realm of rapid-prototyping.
For Jekyll, for keeping me sane.
Acknowledgements

I would like to thank Mark Smith for his invaluable assistance and patience in the writing of this paper.

I would also like to thank Dr. Ross Smith, for his continued guidance and support in all my endeavours in the Wearable Computer Laboratory.

A special thanks also to my Dad, for the drafting and editing he has provided. Any typographical or grammatical errors in this document are my fault, in spite of his assistance.
## Contents

0.1 Abstract .................................................. 1

Acknowledgments ........................................ 3

1 Introduction ............................................. 9
  1.1 Introduction .......................................... 10
    1.1.1 Research Objectives ............................. 10

2 Background ............................................. 11
  2.1 Brief overview of the reasoning for combining Augmented Reality with Tangible User Interfaces ............................................. 12
  2.2 Tangible User Interfaces ................................ 13
  2.3 Augmented Reality .................................... 15
  2.4 From the Head Mounted Display to Spatial Augmented Reality ................. 16
  2.5 Spatial Augmented Reality ........................... 17
  2.6 Tangible Augmented Reality .......................... 19
  2.7 The current prototyping work flow ................. 21
  2.8 Integration of SAR, TAR and the prototyping work flow ................... 22

3 Implementation ......................................... 23
  3.1 Integrating intelligence into Tangible Interface Objects ................. 24
    3.1.1 Development of Intelligent TIO ................. 24
  3.2 Sensors and IntelliTIO .................................. 26
    3.2.1 Powering IntelliTIO ............................ 26

4 Integrating sensors into IntelliTIO .......................... 28
  3.3 Overview ............................................. 29
    3.3.1 Primary goals of the Blob ....................... 29
    3.3.2 Hardware Framework ............................. 29
    3.3.3 Choice of sensors ............................... 30
  3.4 Description of the Blob hardware ....................... 31
    3.4.1 Conceptual overview of the Wasa Board .............. 31
    3.4.2 Accelerometer .................................. 32
3.4.3 Controlling the Wasa Board ................................................. 32
3.4.4 Extending the Wasa Board ................................................. 34
3.5 Description of the Blob software ........................................... 35
3.5.1 Integration into existing SAR systems .................................. 35
3.5.2 Micro-controller integration .............................................. 36
3.6 Blob Development ............................................................... 37
3.6.1 Light sensors ................................................................. 37
3.6.2 LED ............................................................................. 38
3.6.3 Software development ..................................................... 39
3.7 Performance of the Blob ....................................................... 43
3.7.1 Testing the Blob .............................................................. 43
3.7.2 LED indicator design ...................................................... 44
3.7.3 Filtered LDR ................................................................. 44
3.7.4 Multiple LDR ............................................................... 45
3.7.5 Power vs Cost ............................................................... 45
3.7.6 Communication between the SAR system and Blobs ............. 45
3.8 Conclusions ...................................................................... 47

II Intelligent self-powered Tangible Interface Objects 48
3.9 Introduction ................................................................. 49
3.9.1 Blob limitations ........................................................... 49
3.10 RFID Overview ............................................................... 51
3.10.1 RFID Systems ............................................................. 51
3.10.2 RFID Reader to Transponder communication .................... 51
3.11 Classification of Input Controls ........................................... 53
3.11.1 Valuated Input Controls ............................................... 53
3.11.2 Switched Input Controls ............................................... 53
3.11.3 Providing designers with Input Controls .......................... 54
3.11.4 System functionality .................................................... 55
3.12 Choosing input controls for RFID integration ....................... 57
3.12.1 Two-state switched input devices ................................... 57
3.12.2 Valuated input devices ................................................. 57
3.13 RFID Antenna Design ....................................................... 60
3.13.1 Antenna Design .......................................................... 60
3.14 Input control functionality over RFID ................................. 62
3.14.1 Switched input control ............................................... 62
3.14.2 Valuated input control ................................................. 64
3.14.3 Generalizing for input controls ..................................... 64
3.15 Analysis .......................................................... 66
3.15.1 Benefits ............................................................... 66
3.15.2 Drawbacks ............................................................. 67
3.16 Conclusions .............................................................. 68
List of Figures

2.1 The reality-virtual continuum, adapted from [1]. . . . . . . . . . . . . . . 12
2.2 Tangible objects provide a physical representation to digital information,
which can be further supplemented with intangible representations such
as video projections [2]. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

3.1 Conceptual overview of the Wasa Board, version 1.7 . . . . . . . . . . . . 31
3.2 When the Z axis of the accelerometer and gravity are misaligned due to
tilting of the Blob, the X and Y axis of the accelerometer undergo some
acceleration from gravity. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
3.3 Logical layout of the Blob Software . . . . . . . . . . . . . . . . . . . . . 35
3.4 Spectral response of LDR [3] . . . . . . . . . . . . . . . . . . . . . . . . 38
3.5 An RFID transponder. [4] . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
3.6 RFID transponders of the type commonly used in libraries. [5] . . . . . . 52
3.7 Schematic of the Astable Multivibrator [6]. Note that $R_2$ is replaced by
a potentiometer. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
3.8 Single-loop RFID antenna with a gamma-matched network . . . . . . . . 61
3.9 Two variable capacitors and the swamping resistor . . . . . . . . . . . . . 61
3.10 Antenna attached to an antenna analyzer for tuning . . . . . . . . . . . . . 62
3.11 The RFID transponder . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
3.12 Access points attached to the transponders internal antenna . . . . . . . . 63
3.13 Relatively high frequency square-wave, generated by the astable-multivibrator 64
3.14 Relatively low frequency square-wave, generated by the astable-multivibrator.
Note the change in frequency from 3.13 is determined by the changed re-

3.14 resistance of the potentiometer . . . . . . . . . . . . . . . . . . . . . . 65
List of Tables

A.1 Basic Wasa Board AT commands ............................................. 73
A.2 Wasa Board AT commands for reading and writing to the digital and
   analogue pins ................................................................. 74
A.3 Extended Wasa Board AT commands available ............................ 75
Chapter 1

Introduction
1.1 Introduction

Tangible Spatial Augmented Reality can be a valuable tool in the prototyping work flow, allowing quick, cheap prototypes to be developed and explored without requiring the expense, both in time and monetarily terms, of developing fully-functioning prototypes. While Tangible User Interfaces allow a richer immersion and added functionality to the prototype, physical control systems can be better emulated through the utilization of more intelligent Tangible Input Objects. This thesis presents some methods of integrating this intelligence into the Tangible Input Objects.

An additional challenge imposed by the use of Tangible Input Objects is the manipulation of multiple control devices. Purely virtual objects can be collected into groups and manipulated in a multitude of ways which prove difficult to apply to physical objects. Grouping and manipulating coupled virtual and physical input objects provides an additional level of fidelity, enhancing the usefulness of Tangible Spatial Augmented Reality for prototyping purposes.

1.1.1 Research Objectives

This research aims to explore the enhancement of physical input controls commonly used in SAR with a focus on the types of input controls used when utilizing SAR as a design prototyping tool. This research aims to examine increasing the intelligence of such input controls, exploring the types of state that such input controls can encompass, and demonstrate proof-of-concept input controls exhibiting greater intelligence than currently found in TUI objects. This will extend on the work done in [7] to investigate improving the usability, flexibility, functionality and form of physical input controls. The outcome of this research is expected to find that creating more intelligent tangible objects is both possible and useful within the realm of AR-based rapid-prototyping work flows, allowing more natural, flexible work flows for large systems.
Chapter 2

Background
2.1 Brief overview of the reasoning for combining Augmented Reality with Tangible User Interfaces

In 1965, Sutherland revolutionized the computing domain with his vision of a world filled with ubiquitous computing devices. In a room in such a world, he said, humans and computers would interact seamlessly, to the point that “the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal” [8]. Ever since, the field of computer science has strived to achieve this goal and, in so doing, the field of Mixed Reality was created.

Milgram, in his 1994 paper titled “Augmented Reality: A class of displays on the reality-virtuality continuum” [1], placed four positions on the continuum between the real and virtual worlds, as depicted in Figure 2.1. On this continuum, Augmented Reality is defined as the “supplementation of the real world with virtual (computer generated) objects” by Azuma et al. [9]. This supplementation of the real world has numerous applications, and this review will focus on that of prototyping work flows.

In order to interact with this supplementation of the real world, an interface between the real and digital information must be used. In 1997, Ishii described the use of “physical forms that fit seamlessly into a user’s physical environment” [10], which when combined with Augmented Reality, provide a natural, ubiquitous interface between the real and virtual world. In providing this seamless interface, the combining of Augmented Reality with Tangible User Interfaces moves closer towards the world envisioned by Sutherland.
2.2 Tangible User Interfaces

Since the inception of computers, tangible objects have been used for input. For example, in the 1987 paper titled “Designing the Star User Interface”, a precursor to the mouse and keyboard was used [11] to facilitate Human-Computer Interaction (HCI). These two generic input devices are still the most commonplace HCI tools in use today. However, these generic HCI interfaces create restrictions on the interactions possible between humans and the digital world. Ishii [2] says, “We cannot take advantage of our evolved dexterity or utilise our skills in manipulating physical objects” when we use such generic HCI interfaces. Much research is being conducted, delving into the use of natural objects as bridges between the real and virtual worlds. Ishii, in his 1997 paper titled “Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms” described a vision of a world in which human-computer interaction could occur seamlessly, utilising objects already in the users environment to manipulate digital information [10]. By using tangible objects already found in the user’s physical environment as interfaces between the real and virtual worlds, the immersion experienced by the user can be increased, and a more natural, ubiquitous environment can be formed, moving mankind ever closer to the world envisioned by Sutherland [8]. Ishii says that tangible user interfaces provide “tangible representation to digital information”, making “information directly graspable” [2], seen in 2.2.

Research has been applied in moving towards this vision, which is coupled tightly with the integration of digital data with the real world. Indeed, throughout Ishii’s 1997 paper, Augmented Reality is coupled tightly with the applications described [10]. Without this coupling, the usefulness of the Tangible User Interface is diminished, as the object becomes a generic computer input device, much like the mouse and keyboard are today. However, when coupled with augmentation of data onto the physical objects, as described by Ishii, the Tangible User Interface affords a much greater move towards “bridging the gap between the worlds of bits and atoms” [10].

To make greater use, therefore, of Tangible User Interfaces, it is necessary to further develop the technology coupled with it; that of augmentation of digital data onto the real world.
Figure 2.2: Tangible objects provide a physical representation to digital information, which can be further supplemented with intangible representations such as video projections [2].
2.3 Augmented Reality

In 1965, Sutherland stormed the world with his revolutionary vision of The Ultimate Display [8]. He followed, three years later, with the first Virtual Reality (VR) prototype [12], a system that allowed a user to experience a three-dimensional world distinct from that of his natural habitat [12]. His pioneer work created the field of VR and, ever since, researchers have been striving to bridge the gap between real and virtual worlds. However, the early VR systems, and indeed, many of those since, have severe limitations. They limit their users by forcing the use of head-mounted displays, and are detached from the real-world [12] [13]. Augmented Reality seeks to, rather then completely replace the real world, supplement it [13], thus addressing some of these limitations [14].

AR is defined by Azuma et al. as a “supplementation of the real world with virtual (computer generated) objects” [9]. In “Tangible bits: towards seamless interfaces between people, bits and atoms” [10], this idea was further refined, as Ishii put forward the notion that digital bits and real-world atoms should work together, in an integrated, ubiquitous system. AR can be applied to a wide range of fields and applications, and is not limited to visualisations — AR can be applied to sensory inputs for potentially all senses [9]. However, for the majority of applications and, certainly, for the remainder of this review, the focus shall be on visual supplementation of the real world with digital data.
2.4 From the Head Mounted Display to Spatial Augmented Reality

Integrating stimulation for multiple senses is difficult, however, and several AR techniques pose further challenges in immersing the user in an augmented world. Loss of immersion caused by the stimulation method removes the user from the virtual world, rather than connecting them with it. With particular relevance to this review is the challenge presented by the AR Head Mounted Display (HMD). Whilst a HMD is particularly useful in some areas of AR, as described by Thomas and Sandor [15], for example, it poses several constraints onto the user. Bimber and Raskar [16] list several of these constraints, which can be enumerated thus:

- Limited resolution due to the constraints of the technology, both for optical see-through and video see-through displays,
- Field of View constraints,
- Ergonomic constraints, as the right balance between high quality (and more cumbersome) and smaller, lower-resolution displays is sought,
- Difficulty tracking and calibrating the system to the real world,
- Simulator sickness caused by latency and movement in augmented visuals.

In addition to these not insubstantial constraints, it should also be noted that, by necessity, HMD technology limits the collaboration abilities of the users. All users must utilise such equipment in order to participate in collaborative meetings, which must also be planned — the use of equipment by nature limits the ability of a user to spontaneously join such a collaboration effort. This places large constraints on the ability for AR to be utilised in spontaneous, collaborative, rapid design meetings. Whilst HMD technology does have a place, for prototyping, an often collaborative design process, the limitations imposed by the technology do not overcome the benefits provided by it to enhance the work flow.

To overcome some of the constraints posed by pure Augmented Reality, Raskar et al. proposed Spatial Augmented Reality (SAR) [17].
2.5 Spatial Augmented Reality

Spatial Augmented Reality is defined as the “augmentation of physical objects with images integrated directly in the user’s environment, not simply in their visual field” [18]. It is this distinction between augmentation solely in the user’s visual field, and augmentation outside of the users visual field that underpins SAR.

A simple example of this supplementation is the use of a data projector to display information on a screen, thus enhancing the ‘real’ world with the ‘virtual’. However, SAR takes this further by supplementing three-dimensional objects with perspective-correct viewpoints, rather than using a flat screen. SAR achieves this supplementation without hindering the user — ‘science-fiction’ style helmets do not need to be worn, for example, to participate in the SAR environment — and also allow multiple users access to the environment in a collaborative fashion. Each user perceives the environment from the users distinct perspective, as opposed to all users having the same viewpoint, as they would if they viewed a scene on a traditional computer screen.

There are two methods of achieving this integration; either via directly embedding images into the environment (such as through the use of flat panel displays), or through the use of projected images onto real-world objects [18]. This review focuses on the latter.

Projected SAR uses one or more projectors to superimpose light images over real world objects. To achieve this, several challenges must be overcome [18]:

- Calibration between one or more projectors and the real world objects,
- Creation of seamless images when multiple projectors are used, overcoming the issues of disparities when overlapping projections interact.

The second of these challenges is addressed in depth by several authors, including Ramesh et al. [18], Raskar, [19], Bimber [16] and others. Several techniques have been devised, and, as well as allowing seamless images to be projected from multiple sources at differing angular positions relative to the projection target, other techniques to supplement and enhance the immersion have been refined. Arguably, the most important of these is the tracking of real-world objects allowing projected images to keep their orientation and position relative to the real-world object while it is moved [20, 21, 22]. However, others include:

- Projection of shadows and time-of-day lighting effects [21, 22]
- Avoidance of image occlusion [23]

Prototyping with SAR has been successfully implemented numerous times. Owing to the lack of single-user technology, and the increased realism that can be achieved via augmentation of the world without the use of intrusive technology such as a HMD, SAR has distinct advantages over pure AR for this purpose. Marner used SAR for the creation of models from foam sculpting [24], and Porter followed with the use of SAR for prototyping user control systems such as car dashboards [25].
However, by combining the use of Tangible User Interfaces with Augmented Reality or SAR, an even more seamless interaction between the user and the digital world can be achieved. Augmented Reality and Tangible User Interfaces have long been used together in the field of Tangible Augmented Reality.
2.6 Tangible Augmented Reality

Tangible User Interface theories have as a primary focus the idea of seamless integration of real and virtual worlds. This seamless merger of real objects into the digital world lends itself easily to the field of AR. Billinghurst, in “Designing Augmented Reality Interfaces” [26] defines the field of Tangible Augmented Reality (TAR) as a coupling of the TUI and AR domains to form an interface with the following properties:

1. Physical objects are logically connected virtual counterparts,

2. Interaction with the virtual component is achieved through manipulation and interaction with the physical object.

TAR in this sense provides several advantages to AR interaction over and above that of traditional, HCI interfaces such as the mouse and keyboard, as the physical objects used have their own unique properties and attributes that constrain their use [27]. This constraint is an advantage, in that the physical objects are, in that sense, easy to use. However, TAR inherits major disadvantages from TUI in that physical object characteristics are difficult to change, and their use in HCI is not easily apparent, due to an inability to easily determine what digital data or object is tied to the physical object [27].

While this is a comparative disadvantage to the use of TUI and, in particular, to TAR, it can be overcome through a carefully designed user interface, creating an interaction technique that is both robust and lightweight. Seichter et al. demonstrated this in their 2009 work [28]. However, TAR is limited in the same manner as AR – that is, it is traditionally performed using HMD or augmented video-displays. Anabuki [29] identifies some of these limitations in a study using TAR to prototype 3D digital models, saying that “this form did not always satisfy . . . modelling demands”. Some of these drawbacks can be overcome by combining TAR with SAR.

The use of Tangible Augmented Reality provides several advantages over traditional user interface devices, as outlined above. The application of these advantages to the field of SAR, however, provides numerous benefits, in particular over the display technology. Whilst TAR is limited to video-projection or HMD displays, combining TAR techniques with SAR allows the benefits of SAR to mesh with the benefits of TAR. The major benefits gained include:

- Multi-user ‘walk in’ environment — not being tied to HMD technology,
- Relatively inexpensive technology — projectors vs HMD technology,
- More naturally immersive and less obtrusive then traditional AR.

For the field of industrial user interface prototyping, this is particularly useful. While studies have shown that SAR alone is a useful technology for use in prototyping user interface control panels, such as achieved by Porter [25] who developed car dashboard prototypes, the integration of TAR provides several additional advantages to the process.
• Providing a more natural, realistic scenario for prototype testing. Whilst the use of solely virtual buttons can be helpful in the prototyping process, virtual buttons have some drawbacks, in particular the lack of visual depth and the lack of haptic feedback. For prototyping user interface control panels, both these elements are critical — visual depth to allow button occlusion, style and ergonomic issues to be highlighted, and haptic feedback to allow a more realistic prototype to be created. With a tangible object, TAR overcomes these drawbacks.

• Allows for more realistic prototypes to be created, making the prototyping process more useful.

These advantages facilitate enhancements to the prototyping work flow.
2.7 The current prototyping work flow

In today’s industry, prototyping user interfaces, particularly for larger devices, is a process involving multiple iterations:

1. Plan a layout, typically through the use of Computer Aided Design (CAD) software
2. Finalise the design layout (computer view)
3. Create a prototype
4. Test the ergonomics / usability of the prototype

This process dictates that, for every minor change necessitated to the prototype, either:

- A new prototype must be created, an often expensive and time-consuming process
- The minor change is left out of the prototype, leading to misleading and incorrect testing and inadequate designs

In addition to this inherent property of the iterative design process, this iterative process has several downsides, in particular:

- Physical prototypes are expensive and time consuming to manufacture
- Minor flaws in the prototype can not be easily fixed without the creation of a new prototype

Thus, the current rapid prototyping techniques can impede on the design process, rather then facilitating it.
However, the use of TAR techniques in conjunction with SAR is not new, particularly for prototyping. Hisada et al. in “The HYPERREAL Design System” \[30\] describe the merger of SAR and TAR techniques for prototyping objects without the need for creation of physically changed objects. The techniques presented are similar to those used by Bandyopadhyay et al. in “Dynamic Shader Lamps: Painting on Movable Objects” \[20\], in which tracked tangible user interface inputs were used to virtually paint real objects using SAR. It is upon this foundation that Thomas et al. proposed “Glove Based Sensor Support for Dynamic Tangible Buttons in Spatial Augmented Reality Design Environments” \[7\], finding that TSAR was a useful method for prototyping user interface control panels. This study integrated Tangible User Interfaces, Spatial Augmented Reality as well as RFID tracking to form a functioning, flexible prototyping system that could be manipulated and undergo design iterations quickly, and without losing the functionality of it’s components \[7\]. This system moves ever closer to the ideals proposed by Sutherland \[8\] and Ishii \[10\] in integrating virtual and real worlds together. Whilst the study by Thomas et al. does have some limitations, the combination and integration of SAR and TUI allows for the possibility of digital prototyping systems which are greatly improved over current, existing technology. All of the benefits of SAR, combined with the benefits of TAR, eliminate many of the disadvantages of the existing prototyping work flow.
Chapter 3

Implementation
3.1 Integrating intelligence into Tangible Interface Objects

The use of *Tangible objects* in the SAR scene is integral to creating an immersive experience for the user. Within the SAR scene the virtual objects are connected to the physical world through the tangible objects, and the integration of the virtual and physical world relies on the fidelity of the tangible objects. An example of a common, basic SAR environment is the use of a data projector and a projector screen; the *physical* projector screen is augmented with the *virtual* information via the data projector. However, the projector screen is an example of a low-fidelity, ‘dumb’ object. Whilst it is integrated into the scene and augmented with virtual information, it is not utilized as an input to the scene — the interaction between the scene and the object is strictly one-way, from the *virtual* world to the *physical* world. In addition, the immersion provided by the projector screen is limited to visual information; it does not provide haptic, tactile or other sensory information.

Transitioning from a ‘dumb’ tangible object to an object which can be used as an *input source* provides a large jump in the immersion of a SAR scene. To extend the above example, *smart boards* can be utilized as both a *projector screen* and an *input source*; the user can both receive information from the SAR system via the augmented data display, and can input data into the scene through the use of whiteboard pens etc. The user can further overlay the virtual data — the projected information — with physically inscribed data. This allows much deeper interaction between the user and the SAR scene, as the *tangible object* has been converted into a *tangible input object*, or TIO.

There are numerous examples of TIO being used in SAR scenes to augment the users ability to interact. The ‘virtual spray paint’ presented by Mass et al. [31], for example, uses a tracked TIO to virtually ‘spray paint’ objects within the SAR scene. Another example used plastic, non-functioning buttons in conjunction with an RFID glove to emulate button functionality [7]. However, both these system still suffer from some deficiencies. In particular, the functionality provided by the TIO is *emulated*; the apparent functionality is provided only through the augmented data stream. The SAR system must track the spatial position and state of the TIO, interpreting interactions between the user and the TIO to emulate the TIO functionality, such as a button press, or color-change.

The next logical step in emulating physical functionality in the Tangible Input Objects, such as two-state switches, is to provide the TIO themselves with the knowledge of state and position. In so doing, the TIO can communicate with the SAR system to inform the system of position or state change, enhancing the SAR system as a whole and providing additional, hitherto limited, functionality to the user.

3.1.1 Development of Intelligent TIO

The development of ‘intelligent’ Tangible Input Objects can aid the creation of immersive, functional scenes, allowing designers to more easily utilize SAR for rapid prototyping work flows.

24
1. Provide additional tactile, haptic and sensory information.

2. Increase the immersion of the SAR scene

3. Improve the fidelity and realism of the TIO in the SAR scene.

By granting the TIO self-awareness — that is, knowledge of various state information, such as spatial position or switch state (‘on’ or ‘off’) — the TIO can communicate both with the user and with the SAR system. For example, a self-aware TIO containing a two-state switch could communicate with the SAR system when its state changes, eliminating the need for the SAR system to actively track and check the state of the TIO. Another example may be a TIO that tracks changes in spatial position; such a TIO could communicate with the SAR system, minimising some of the issues caused by visual-tracking occlusions.

The immersion of the SAR scene can be greatly enhanced by such ‘intelligent’ TIO. Minimising complications such as visual-tracking based occlusions, for example, can greatly enhance the SAR scene from the users point of view. Furthermore, state-aware TIO could provide indications to the user of their state. This would be particularly useful in prototyping applications, such as indicator lights for power switches. While such functionality can be emulated successfully via existing SAR systems, the use of TIO to achieve this functionality may increase the speed at which prototypes can be developed if, for example, such ‘intelligent’ TIO are simple to use.

Tangible Input Objects which are self-aware are deemed ‘IntelliTIO’. 


3.2 Sensors and IntelliTIO

Integrating sensors into Tangible Input Objects provides numerous benefits. One critical aspect of the TIO used in prototyping is the self-awareness of state. This self-awareness allows the SAR system to determine, for example, whether a TIO is ‘on’ or ‘off’, or the position of a rotary dial or bidirectional slider, enabling the SAR scene to provide functionality to the user that is otherwise challenging to emulate. In the field of rapid prototyping, providing TIO which have differing physical states allows the designer to prototype control systems with a deeper immersion than is possible through the use of pure virtual state. The designer is able to turn a dial, press a button or switch a switch, and have the SAR system respond.

While this is possible using purely-virtual emulated functionality — for example, determining if a button has been pushed by tracking a users fingers — it is exponentially more challenging, and brings a host of difficulties. In particular, the need for additional complexity in the SAR system, and the greater effect of tracking occlusions almost entirely negates the benefits of the emulated state, particularly in a rapid prototyping environment. For these reasons, the following sections explore several methods of creating state-aware TIO using a variety of sensors.

Sensors are well used in embedded systems, and can be used to measure state for a wide range of natural phenomenon. Some of the natural phenomenon commonly measured by sensors include electromagnetic radiation, pressure and temperature. Within the realm of Human-Computer Interaction, electromagnetic radiation in particular is well-used. Wireless communication protocols utilize a wide range of the electromagnetic spectrum, including visible light, radio waves and microwaves. In addition, within SAR, visible light is singularly important, but numerous examples of natural phenomenon, such as pressure in the form of sound waves, have been used in the field of Augmented Reality.

3.2.1 Powering IntelliTIO

Non-powered TIO can provide some intelligent functions, such as mechanical state-awareness. Simple two-state switches would be an example of this. However, for any sufficiently complex notion of ‘intelligence’, electric circuits, and a method of powering these circuits, are required.

Intelligent TIO can be classified in numerous ways. The following chapters, however, will delimit between differing methods of powering IntelliTIO. The method via which the IntelliTIO is powered has a direct impact on both the types of sensors which can be used by the IntelliTIO, and also on the usability of the IntelliTIO within the SAR scene.

Power sources can either be internal or external. For the purposes of this document, an ‘internal’ power sources refer to traditional power sources, such as batteries, while ‘external’ power sources refers to power sources external to the object. Such ‘external’ power sources may include power transmitted over communication wires, such as that found in the Universal Serial Bus (USB) protocol, which can power the devices connected to it.
Using a ‘traditional’ power source, such as a battery or physical attachment via cable, provides several benefits to the TIO. There are few limitations on the amount of power available to the TIO, imposing fewer limitations on the number or complexity of the sensors utilized. In addition, the useful communication range of the TIO may be improved when more power is available to it. The use of batteries also imposes a monetary cost to the TIO which may effect the feasibility of the technology.

Energy harvesting, the process of capturing energy from an external source, can be utilized for a wide range of energy sources, including solar, thermal and electromagnetic radiation. A prime example already in use in several SAR systems is Radio-Frequency IDentification, (RFID), some examples of which can be seen previous work I have contributed to [7, 32].

Energy harvesting provides some unique benefits to TIO, in particular with regards to maintenance and portability — finite resources such as a battery require either charging or replacing, for example. RFID and other energy harvesting technologies can allow the integration of electronic circuits into the TIO while eliminating the need for the maintenance of the TIO.

Furthermore, as many energy-harvesting techniques alleviate the need for attachment via cables to the SAR system, the energy-harvesting TIO may have a greater flexibility than some powered counterparts.

However, energy harvesting comes at a cost; in particular, the size of the power draw of the TIO is quite limited, as the amount of energy available for scavenging at a given point in time is itself limited.

In Parts I and II, we develop both traditionally-powered and energy-harvesting IntelliTIO, as demonstrations of both techniques. The intended application of the TIO, as well as the number and type of sensors to be used, will obviously have a strong impact on the feasibility of either traditionally-powering, or the use of energy-harvesting techniques.

In addition, creating intelligent TIO from ‘traditional’ input controls is explored in chapter II, demonstrating the type of input controls used extensively when prototyping control systems.
Part I

Integrating sensors into IntelliTIO
3.3 Overview

‘Intelligent’ Tangible Interface Objects, or IntelliTIO, can greatly enhance the immersion and provide capabilities to the SAR system that are difficult to emulate or attain through the use of non-intelligent objects. Providing the TIO with awareness of self, the localized environment, and the SAR system as a whole is achieved through the integration of sensors with the TIO. It is through these sensors that the TIO gains knowledge of both itself and its environment; the ability to communicate with the SAR system is also gained through the use of sensors to both transmit and receive data.

To demonstrate some of the functionality achievable through the integration of sensors and TIO, a proof-of-concept IntelliTIO was developed, incorporating sensors providing information about the change in spatial position and local environmental lighting. This IntelliTIO is henceforth deemed a ‘Blob’.

3.3.1 Primary goals of the Blob

In order to facilitate the use of the IntelliTIO within a SAR scene, and to allow future reuse and extension of any IntelliTIO, several key aspects must be addressed:

Cost: The IntelliTIO must be cheap to manufacture.

Extensibility: Both the hardware and the software must be flexible and extensible to allow different sensors to be added, and to facilitate integration into the SAR software framework.

Power draw: The IntelliTIO must have as low a power draw as practical, to reduce either the maintenance if the IntelliTIO uses an internal power source, or the required infrastructure if the IntelliTIO utilizes an external power source.

The proof-of-concept Blob meets these criterion, and demonstrates that sensors and Tangible Input Objects can be merged to create TIO that are aware of their own state and the state of the local environment. The Blob demonstrates that a low-power, sensor-integrated TIO is feasible, cheap to produce, and achievable with current technologies.

The development of the Blob focused on both the hardware development and the software integration with existing SAR systems. Both the hardware and the software are modular in nature, which allows ease-of-development and seamless integration into existing SAR software libraries.

3.3.2 Hardware Framework

Rather than developing a hardware platform from scratch, the Blob extends a Wasa Board, an open hardware, software embedded controller board [33]. The Wasa Board is designed to facilitate the extension of existing hardware by enabling the simple addition of sensors. The architecture of the Wasa Board is specifically designed to allow additional sensors to be easily integrated to the Board, which provides ease-of-development for
the proof-of-concept Blob and also caters for future prototypes to extend the sensory capabilities of the Blob.

It should be noted that, although the Wasa Board and, by extension, the ‘Blob’, are designed to be low-power, the ‘Blob’ is not power-scavenging; instead, it is powered via the serial connection.

3.3.3 Choice of sensors

The Blob is designed to be aware of changes in its spatial position, and to have the ability to communicate with both the user and the SAR system. To that end, the sensors used in the proof-of-concept Blob include an accelerometer, Light-Dependant Resistors (LDR), and Light-Emitting Diodes (LED).
3.4 Description of the Blob hardware

This section details the conceptual, logical and physical layout of the Blob hardware. As the framework the Blob is built from is a Wasa Board, this section first outlines the Wasa Board functionality, then describes the extensions made to the version 1.7 Wasa Board to create the Blob. Finally, this section details the hardware integration necessary to build the Blob prototype.

3.4.1 Conceptual overview of the Wasa Board

The Wasa Board is designed to be a modular framework, allowing sensors to easily be attached to and controlled from a micro-controller. To that end, a conceptual overview of the Wasa Board can be seen in figure 3.1.

The sensors on a generic version 1.7 Wasa Board include two analogue sensors, a light-dependant resistor and temperature sensor, and a digital accelerometer.

Additional sensors can be attached to the expansion connectors.

Key Blob sensors

The Blob aims to communicate several key characteristics with the SAR system. It does this through the combination of several sensors, outlined briefly below.
Changes in Spatial Position  Spatial position changes are read through the accelerometer and communicated to the SAR system through the light emitting diodes.

localized environmental lighting  The SAR system communicates with the Blob through changes in the color of light projected onto the Blob within the SAR scene. This is read through the light-dependant resistors.

3.4.2 Accelerometer

The accelerometer used on the Blob provides 6 digital bits of resolution in relative X, Y and Z directions. Although finer-grained resolution is possible, this is sufficient resolution to distinguish the acceleration values that typically occur within a SAR scene, such as when the Blob is picked up, rotated or moved. The Blob aims to complement the SAR tracking system, rather than replace it fully. To that end, the resolution provided by the accelerometer used is sufficient.

The effects of tilt

One important characteristic of the accelerometer is the relative measure of the X, Y and Z axis. If any tilt is applied to the board, the axis of the board disarrange from the axis naturally understood by humans, where Z is upwards. For example, if the board flipped upside down, and the board is picked up, from the accelerometers point of view, the board has moved downwards.

This has a large impact on the perceived acceleration when the board is tilted. When the X and Y axis are orthogonal to the effects of gravity (the Z axis points either directly up or directly down), the X and Y axis undergo no acceleration from gravity. However, if this orthogonal relationship is disarranged, there is tangential acceleration from gravity on the X or Y axis. This can be seen in figure 3.2. In the figure, the accelerometers X and Z axis are shown (Y is neglected for ease-of-drawing — the same principle is applied to the Y axis, however). The accelerometers Z axis is at an angle to gravity and, as a result, a component of the force of gravity is applied to the X axis. The component force on the X axis can be calculated using the formula $x = 9.8 \sin \alpha$. While the accelerometer is tilted, the Z axis will undergo less acceleration from gravity, given by the formula $z = 9.8 \cos \alpha$. In both formulas, $\alpha$ is the angle between the accelerometers Z axis and the direction of force from gravity.

3.4.3 Controlling the Wasa Board

The board is controlled through a subset of the Hayes command set [34, 35], a command language developed for use in modems. The command set consists of text strings which are combined into commands. Because many of the commands begin with the string AT, the Hayes command set is also known as the ‘AT commands’. There are numerous command operations for the Wasa Board, including polling individual sensors, data streaming operations, and data output operations. The command set used by the Wasa Board can be extended to allow user-specific operations. The full set of Wasa Board
Figure 3.2: When the $Z$ axis of the accelerometer and gravity are misaligned due to tilting of the Blob, the $X$ and $Y$ axis of the accelerometer undergo some acceleration from gravity.
commands can be seen in appendix A. In addition to providing hardware extension support 3.4.4, the Hayes command set usable by the Wasa Board can be expanded if necessary, including the creation of custom Hayes commands.

3.4.4 Extending the Wasa Board

The Wasa Board is extended through the addition of several sensors to create the proof-of-concept Blob. The two types of sensors chosen for the Blob are Light-Dependant Resistors (LDR) and Light-Emitting Diodes (LED). The LED are attached to the expansion connectors as shown in figure 3.1. An unmodified Wasa Board contains a singular LDR. To complement this, an additional LDR was attached in place of the temperature sensor that the Wasa Board is built with. Both of the LDR are thus included in the ‘analogue sensors’ as seen in figure 3.1.

The Blob demonstrates the ability to react to color-specific light frequencies. To provide this functionality, a opaque filter was constructed from acrylic plastic and attached to the LDR. When white light is shone at the filter, some light is absorbed by the filter, while the frequency of light matching the color of the filter passes through to the LDR. In this manner, the LDR becomes color-sensitive.

Control of the additional sensors is achieved through the use of existing Hayes command set.
3.5 Description of the Blob software

The logical layout of the software portion of the Blob can be seen in figure 3.3. The software comprises of four modules:

- Communications
- Blob interface
- Application logic
- Application interface

The software is designed to be modular, with each module having specific, defined roles and responsibilities. It is written in C++, using the GNU GCC version 4.7 compiler on Mac OS X 10.7. This allows the software to be integrated easily into existing SAR systems. In addition, it allows the communication medium between the software and the Blob to be easily changed, by swapping the ‘Communication’ module. This allows future implementations of the Blob to integrate different communications mediums, such as RFID, Zigbee, or Ethernet easily.

3.5.1 Integration into existing SAR systems

The ‘Application interface’ module is responsible for communication between the Blob software and the SAR system software, by providing an Application Programming In-
terface, or API. This API allows the SAR system to utilize the Blob from a high level, without the need to integrate low-level functionality in the SAR system. This modularity allows for ease-of-integration between the Blob software and the SAR system software.

3.5.2 Micro-controller integration

The Blob is capable of running the Blob logic directly on the micro-controller, as opposed to the computer-control via serial connection utilized here.
3.6 Blob Development

The development of the Blob focused on three main areas: frequency-specific LDR, LED communication, and the software.

3.6.1 Light sensors

The two Light-Dependant Resistors on the Blob are generic LDR; they react to the visible light spectrum. However, the Blob aims to demonstrate frequency-specific LDR, while maintaining a low overall cost for the proof-of-concept. For this reason, rather than purchasing an off-the-shelf frequency-specific LDR, a color-filter is used to inexpensively convert the LDR into frequency-dependant LDR.

Filtered LDR

The provision of color-dependent LDR values allows the SAR system to communicate with the Blob, through the use of coloured light. SAR systems incorporate projected light as a fundamental basis for their operation. By leveraging the existing technology of projected light as the system-Blob communication medium, the Blob can be integrated into the SAR system without the need for development of additional subsystems, such as Zigbee- or RFID-based modules. This cuts back on both the additional hardware and software needs for the SAR system.

Color-dependent LDR is achieved by placing a filter between the light source and the LDR, to strip out unwanted light frequencies. The filter absorbs all frequencies of light except that which matches the color of the filter, thus allowing only the same frequency light as the color of the filter through to impinge the LDR. The LDR then reacts to the light, allowing the Blob to determine the intensity of a specific colored light.

Spectral Responsiveness

It should be noted that Light-Dependent Resistors are responsive to a spectrum of frequencies. An example can be seen depicted in figure 3.4, although the exact spectral response will be different for each LDR. This spectral responsiveness means that a specific LDR may perform poorly when detecting a specific frequency at one end of the visible light spectrum, while performing well when detecting a different frequency. However, while this should certainly be taken into consideration when developing a fine-grained frequency-specific LDR, for the proof-of-concept Blob a generic LDR was deemed suitable for use in the filtered LDR.

Filter materials

During the development of the Blob, several different filter materials were tested, including:

- Colored plastic, of the type that shopping bags are made from.
Colored plastic, of the type that soft drink bottles are made from.

Acrylic plastic

All tested materials were subject to several limitations:

- ‘Frequency bleed’: the materials do not absorb all frequencies, letting a large range of light frequencies through.

- ‘Reduced light intensity post-filter’: the light that passed through the filter was greatly reduced in intensity, even when the frequency was a close match to the color of the filter material.

‘Frequency bleed’ was found to be particularly detrimental with red shopping bag plastic. Testing the red filter with a blue LED source light, it was seen that a large amount of light passed through the filter; however, due to the poor quality of the materials, a large amount of frequency bleed is seen with all tested filters.

3.6.2 LED

To facilitate Blob-SAR system communication, several LEDs were attached to the Blob. These LEDs provide a medium which can be read by the SAR system, which typically utilizes cameras to track objects in the SAR scene. Two LEDs were attached to the Blob. The colours used were green and red, chosen due to the relatively large difference
in frequency. Because the filter on the filtered LDR is yellow, a yellow LED was not used, although due to frequency bleeding the filtered LDR responds to the LEDs. To minimise this effect, the LEDs were attached to the Blob with wire, allowing some distance between the LDRs and the LEDs.

By switching the LEDs on and off, the Blob can communicate with the SAR system. A minimum time for each message is determined by the SAR system camera frames per second; however, the Blob operates at 64Hz, thus setting a minimum LED on/off period of $\frac{1}{64}$ of a second. A basic message could be sent to the SAR system by switching a LED on or off; the project demonstrates the red LED switching on when the blob is lit with yellow light, and the green LED switching on when the Blob undergoes acceleration. A series of on/off combinations could be utilized to send more complex messages to the SAR system.

For future implementations of the Blob, an infra-red LED could be used, to provide a communications medium that is invisible to the human eye, while still visible to the SAR system cameras. This would have the secondary benefit of limiting the detraction to the SAR scene that a visible-spectrum LED imposes to the users of the SAR system.

3.6.3 Software development

The software developed for the Blob system utilizes the streaming capabilities of the Wasa board, parsing the string for tokens of interest, and acting on the tokens when flags are set. The main loop in pseudo code can be seen in appendix [B].

Wasa Board streaming

The Wasa Board has the capability to stream various data points over the serial interface, at a rate specified by the user. The stream is delimited by various character combinations. For example:

| AXL | Denotes the accelerometer value section of the stream. |
| ANL | Denotes the analogue section of the stream. |

Accelerometer values

The acceleration indicator LED is activated if the acceleration values in the token stream are above a threshold, set during the program initialization. The process for obtaining the threshold are as follows:

1. **Take samples:**
   During the `init` phase, the program gathers information over 15 time-units. The token stream is parsed and values after `AXL:` are extracted and converted into three integers, `accel_x`, `accel_y` and `accel_z`. These values are then pushed into a `std::vector<int>`.
2. **Average samples:**
The tuple values are summed and averaged to determine an average acceleration.

3. **Determine threshold:**
The threshold values for ‘[x,y,z]’ acceleration are set as the average ‘[x,y,z]’ + 3. This provides a threshold over almost all of the natural variation seen by in the at-rest accelerometer.

It is assumed that the Blob is at rest during the ‘init’ phase. However, sampling of the accelerometer data is necessary even if the accelerometer is not undergoing acceleration, as the accelerometer is an analogue value subject to noise and several unavoidable sources of error converted to a digital value. Thus, the accelerometer data fluctuates, and the token stream the Blob provides does not have a stable acceleration reading. By averaging the 15 samples, a better estimate of the ‘at rest’ accelerometer values of the Blob can be taken.

Once the threshold ‘[x,y,z]’ values have been set, the threshold is compared against the returned accelerometer values and, if exceeded, an **acceleration occurring** flag is set. If the threshold is not exceeded but the flag is set, the flag is unset.

The returned acceleration values are pushed into a FIFO queue of size 15, which is summed and averaged each tick, to update the threshold. By using an average acceleration value, rather than the per-tick acceleration values, the effect of noise sources, particularly board tilt experienced while a user moves the Blob, as described in [3.4.2](#), are drastically reduced.

One issue that occurs from the use of threshold values is that it is possible to move the Blob without activating the indicator light, by keeping the acceleration below the threshold value. While this challenge could be minimised by a higher degree of resolution in the accelerometer, or an accelerometer with a higher sensitivity, with the currently-used Blob hardware it is difficult to solve. Minimisation through software alone, while beneficial, does not completely remove the issue — lower threshold value has the side-effect of triggering the indicator light when the Blob is not undergoing acceleration.

**Conversion of streamed data to integers**

Converting the streamed data string tokens to integers is performed in the following method:

1. Create a **char buffer**.

2. Loop over the relevant section of the token stream:
   - If the encountered **char** is ‘-’, set a flag **isNegative**.
   - If the encountered **char** is ‘[0--9]’, add the char to the **buffer**.
   - If the encountered **char** is ‘,’ break from the loop.

3. Convert the **buffer** to an **integer**:

40
int number;
for i in bufferlen do
    number += ((int) buffer[i] - 48) * (pow(10, bufferlen-1-i));
done;
if isNegative number=number*-1;

The char is converted to an int and subtracted from the ASCII numeric position (ASCII character ‘0’ has integer value 48), then multiplied by the correct decimal column numerator. Finally, if `isNegative` is set, the sign of the number is switched.

LDR values

The LDR values are parsed in a similar fashion to the accelerometer values. During the ‘init’ phase a threshold value is generated for both LDR, by taking 15 samples and averaging the results. This smooths the effect of flicker and other noise sources from ambient and direct lighting on the LDR. Parsing and conversion of the streamed string to integers uses the same function as that shown above.

When a LDR threshold value is exceeded, a LDR on event is triggered within the software, setting a boolean flag for the duration that the LDR threshold value is exceeded. There is a flag for each of the on-board LDRs. This flag is used to trigger events — in the developed Blob, a LED is enabled when the flag is set, notifying the user of the high LDR value. This flag can also be used in timed events, allowing pulses of light to be used as signals from the SAR system to the Blob. This enables a much wider range of communication than only using individual light frequencies would allow.

Threshold values for the LDR are set as percentages of the average sampled light. For the non-filtered LDR, a threshold of 10% is suitable to detect an LED directed at the LDR.

For the filtered LDR, LDR-2, the threshold value is set at 2%. However, this threshold value is subject to some complications:

- Ambient light, particularly fluorescent bulb flicker, can be sufficient to trigger an LDR-2 on event.
- The threshold is not high enough to prevent an LDR-2 on event from triggering when non-yellow light is directed at the LDR; in particular, red light triggers an event.

However, a higher threshold value provides fewer benefits:

- Under testing, a higher threshold value did not guarantee the triggering of an LDR-2 on event when yellow light was used.
- Higher threshold light was still subject to false-positive event triggers when under non-yellow light, in particular red frequencies.
In testing, blue LED light was seen at a value of approximately 1% higher than ambient lighting; the 2% threshold value is thus sufficient to distinguish between blue and yellow LED light. Yellow light, which gives LDR values approximately 4–5% higher than the threshold, triggers the LDR-2 on event, whilst blue light does not.

An additional complication encountered whilst developing the filtered LDR threshold is the unstable nature of the ambient light with respect to the LDR values. Natural variance of up to a percentage are visible whilst streaming the LDR values; this is seen while the Blob is under ambient light or directly under specific-frequency lighting. This unstable baseline makes choosing a suitable threshold value difficult, as the variance occurs even when under the signaling LEDs. This is perhaps due to the analogue nature of the LDR, combined with flickering or minute changes in ambient lighting. The chosen thresholds, while performing well, could perhaps be improved, although the performance improvement gained through fine tuning the thresholds would be vastly out-weighed by that of improving the filter material.

Indicator LEDs

The indicator lights are attached to the GPIO0 and GPIO7 pins on the Blob. During the init phase these pins are set as digital outputs, and are switched on and off by setting the output to 0 or 1.

The indicator lights are activated and deactivated through the use of flags, set when events are triggered. Examples of events include the LDR and LDR-2 binary events, triggered when the respective LDR values are above the respective LDR-threshold values.

Because the Blob utilizes constant streaming, toggling the indicator LEDs causes the streaming to pause. This causes a slight disruption in the streaming data; however, in future versions of the Blob the code can be run directly on the microprocessor, rather than on a serially attached system, which will remove this complication.
3.7 Performance of the Blob

While some aspects of the Blob function reasonably well, others do not, due to inadequacies in the design. Following is a brief discussion of the tests undertaken for the Blob, and a discussion of various performance metrics.

3.7.1 Testing the Blob

Several tests were conducted in order to verify that the functions of the Blob worked. It should be noted that the tests were centered around the Blob as a proof-of-concept system — testing revolves around verification of functionality, rather than an in-depth dissection of performance. Extensive, user-based studies are one area for future research in this problem domain.

LED indicators

The LED indicators were tested to verify both hardware and software functionality. The LED indicators were tested through manual triggering through the Blob software.

Light Dependant Resistors

There were several steps involved in testing the LDR. Firstly, both the filtered-LDR and the non-filtered LDR threshold was established by placing the Blob in ambient light and taking several thousand samples through each LDR. The mean of the samples was then used as the baseline for ambient light.

The second step was to determine the threshold percentile of the baseline value. The highest value in the baseline samples was converted to a percentile, which was used as the initial threshold value. The Blob was placed in ambient light, and two LEDs, one yellow and one red, were placed at a distance of approximately 40cm above the Blob. These LEDs act as the signaling LEDs within the SAR scene.

The effectiveness of the filter on the filtered-LDR was tested by first turning on the red LED and sampling the filtered-LDR. If the yellow filter was near-perfect, and the red LED emitted a single frequency of light, then the average sample taken from the filtered-LDR should be the same as the average from the ambient light samples, as almost no light should pass through the filter from the red LED. Testing revealed that either the filter allows a range of frequencies through, or the red LED emits a range of frequencies, or a combination of the two — the average sample taken while under the red LED was significantly higher than the filtered LDR. For this reason, a blue LED was substituted for the signaling LED, as blue is further away from yellow on the visible light spectrum than red is.

Testing the blue LED in the same fashion showed an increase of approximately 1% over ambient light. This value was used to set the filtered-LDR threshold to 2%. When using the yellow signaling LED, the average sample increased approximately 4%, significantly over the threshold. Testing thus proved that the yellow and blue signaling LEDs could be used in conjunction with the two LDRs. Combinations of individual LDR
events can be triggered by using various signaling combinations, although it should be noted that the yellow signaling LED did trigger the non-filtered LDR event in addition to the filtered-LDR event. Future versions of the Blob could incorporate multiple filters of differing frequency to allow for more fine-grained control in addition to, or in place of, the general, non-filtered LDR.

**Accelerometer**

3.7.2 LED indicator design

Use of the LED indicators performs reasonably well. The Blob signaling capabilities respond well to event triggers, such as acceleration. However, due to the design (constant data streaming from the Blob), switching the LED indicators has the effect of momentarily halting the streaming. Testing determined the average time to send the AT command to turn on or off the indicator takes approximately 15ms, which is approximately a single time-unit for the Blobs default streaming rate. (The default streaming rate is 64Hz; one ‘tick’ takes $\frac{1}{64}$ of a second). While this does not noticeably reduce performance, it should be noted that, if the indicators are toggled multiple times each tick the effective streaming rate can be drastically reduced. This would be amplified by the addition of multiple LED indicators, or other read-write sensors, but would be mitigated by the movement of the control-code from the serially-attached CPU to the on-board microprocessor of the Blob.

3.7.3 Filtered LDR

The filtered LDR on the Blob do not perform well. In particular, there is a large amount of ‘frequency bleed’, which causes the filtered LDR to trigger the LDR-2 on event in the presence of non-yellow light; red and green frequencies are particularly apt for bleeding through the filter. The two most obvious sources of error accounting for this: the filter, which does not perfectly inhibit non-yellow light, and the intensity of the differing LEDs, which may not be constant. In addition, the underlying light-sensor is responsive to a range of frequencies; the response is not flat across all visible light frequencies, and this will also affect the performance of the frequency-specific light-sensor.

Future versions of the Blob should investigate the use of color-specific sensors; several products exist which would fill this role. However, one area that the filtered LDR does perform well is the power draw; the Blob does not require much power, and the addition of other, better performing, color sensors may mitigate this.

Testing the filtered LDR revealed that, provided the light sources are chosen with care to avoid color frequency bleed into the filtered frequencies, the filtered LDR does distinguish between yellow and non-yellow light sources. However, when the filtered frequencies and light source frequencies overlap, the filtered LDR does not manage to distinguish between the yellow and non-yellow light sources — blue LEDs during testing did not trigger the LDR-2 on event, while red LEDs did. In this regard, then, the Blob performs well, although it should be noted that integrating the Blob into a typical SAR
system, which utilizes color over the entire visible spectrum, will lead to numerous false-positive event triggers. One factor which would greatly improve the performance in this area is the inclusion of better filters. Precise filters with known characteristics would provide better performance, less color bleed and allow for frequency-specific signaling lights, rather than general signaling (general “blue” and “yellow”) colors. This may increase the cost of the Blob, however.

3.7.4 Multiple LDR

The two LDR are used together in an effort to overcome the limitations of the filtered LDR. A comparison is made between the filtered LDR and the unfiltered LDR; if both LDR have low readings, it is assumed that the board is under yellow light. If, however, the filtered LDR has a higher reading and the unfiltered LDR has a lower one, then it is assumed that the board is under non-yellow light. (The higher LDR readings indicate a lower light intensity). In practice, this does not perform very well, for several reasons:

- The filtered LDR has, as mentioned above, considerable frequency bleed; it reacts to a wide range of frequencies.
- Testing the two sensors proves to be difficult; owing to the natural variance in readings, and the naturally different intensities of light that reaches the filtered LDR compared to the unfiltered LDR (due to the filter), it is difficult to obtain a consistent measurement when other variables (type, distance and orientation of the light used, ambient room light) are kept as consistent as possible.

Despite these limitations, the Blob is able to distinguish successfully during testing between some non-yellow and yellow frequencies; as a prototype, the Blobs performance proves that, at least to a limited degree, the use of specific light frequencies as a communication medium between SAR systems and TIO is possible.

3.7.5 Power vs Cost

The Blob is designed from a Wasa Board, version 1.7; the modifications to the board are limited, and the Blob prototype can be cut down with the exclusion of pins and sensors that are unused. The Wasa Board is designed to have a low power draw, a characteristic shared by the Blob; cutting down future prototypes would enhance this capability. In addition, the Wasa Board is reasonably cheap to produce; the reduction of the design would drop the cost further, also reducing the form-factor, allowing for flexibility in TIO design.

3.7.6 Communication between the SAR system and Blobs

The multi-direction SAR system to Blob communication system uses both the LDR and the indicator LEDs. Both systems have the ability to detect colored light (read messages) and display light (write messages). This allows the SAR system to interact on a richer
scale than is possible with a traditional ‘passive’ TIO. In future versions of the Blob, other sensors, such as toggle switches, temperature sensors and pressure sensors can be integrated into the Blob and their data sent to or queried from the SAR system. This allows a greater interaction between the TIO and the SAR system, enabling a more immersive SAR scene to be created.
3.8 Conclusions

The prototype Blob meets the criterion specified in §3.3 demonstrating a proof-of-concept sensor-based TIO that is inexpensive, flexible and extensible.

Whilst the prototype developed demonstrates the use of multiple sensors and TIO — SAR communication, there are several areas where the prototype could be improved on, and there are several aspects of the prototype which are not adequate for use in SAR prototyping.

The prototype Blob does not function well in the area of frequency-specific LDR; future implementations should re-design the filter with a different material, or examine the possibility of integrating existing color-specific sensors. This may drive the cost of the Blob up, but would provide more functionality.

The Blob has a low power draw, an ideal characteristic for a TIO for use in SAR. The functionality displayed by the accelerometer indicator demonstrates that sensor-based TIO can provide additional state or positional information to the SAR system.

The biggest change that a future Blob implementation should exhibit is the integration of the program code onto the microprocessor, allowing untethered Blob use, as opposed to relying on a cabled serial connection.

Overall, the Blob demonstrates that sensor-based TIO are a viable and possibly valuable addition to traditional SAR prototyping environments, and provides a platform upon which future TIO development can occur.
Part II

Intelligent self-powered Tangible Interface Objects
3.9 Introduction

Providing the TIO with state-awareness can be achieved in a multitude of methods. An example has been given with the Blob system in section 3.3. However, while the Blob provides several important benefits to the SAR system, it imposes several limitations on the SAR system which are not ideal. In particular, the relatively heavy power draw, which would necessitate a battery-based power source in order to allow for flexible, dynamically movable IntelliTIO within the SAR scene. This limitation imposes an additional maintenance expense on the designer, raising the cost of utilizing the IntelliTIO. This expense is exasperated if numerous Blobs are integrated into the scene.

An outline of this chapter is as follows.

1. I present the limitations of the IntelliTIO presented in 3.3 and outline a method to improve on this limitation 3.9.1 using RFID.

2. In 3.10 I present a brief overview of RFID terms and technology.

3. I classify input controls into two broad categories in 3.11.

4. In 3.12 I detail the creation of two RFID-enabled input controls.

5. In 3.14 I outline the completed IntelliTIO, and their use within the SAR system.

6. Finally, I present an analysis of this work in 3.15 along with avenues for future exploration.

3.9.1 Blob limitations

The major limitation of the presented Blob system is the additional expense imposed by the power draw of the Blob. This limitation can be solved in one of three ways:

1. Internal power source, such as a battery

2. External power source (traditional)

3. Opportunistic power source (power-scavenging)

Of these three solutions, both 1 and 2 are less than ideal. An internal power source imposes an additional maintenance cost on the user of the SAR system, violating the goal of producing an inexpensive TIO. A traditional external power source requires a power-cable of some description, which limits the flexibility of the user to dynamically move the TIO whilst in the SAR scene and, in addition, limits the number of TIO that can be used in a single SAR scene concurrently.

The third option presented does not have either limitation. The TIO ‘scavenges’ power from the environment without needing a tethering power-cable, combining the flexibility of an internal power-source without having an additional maintenance expense. This provides several benefits to the designer.
Thus this chapter will explore the third option, the use of a power-scavenging external power source for IntelliTIO, by integrating Radio Frequency IDentification with state-aware TIO. RFID is a power-scavenging technology which provides an efficient method of identifying individual TIO objects within a SAR scene. It has been successfully used in several SAR-related Tangible User Interface studies [7]. In addition, this chapter will explore a more traditional set of state-aware TIO, including two-state switches and rotary dials, demonstrating the usability of the functionality, and will also discuss future implementations and improvements for the technology.
3.10 RFID Overview

The integration of Radio Frequency IDentification (RFID) with TIO provides several key benefits. RFID provides both a communication medium between the TIO and the SAR system, as well as providing a means of scavenging power, allowing the TIO to function without the use of a ‘traditional’ power source such as a battery.

By utilizing RFID as a communication medium, the TIO can transmit state information to the SAR system. This allows for the creation of several conventional controls, including switched and valuated input devices.

For the purposes of this document, it is beneficial for the reader to have a brief understanding of the RFID protocol. The following section explains basic RFID principles and operation.

3.10.1 RFID Systems

A Radio-Frequency IDentification system is made of two parts, a ‘transponder’ and a ‘control device’, or ‘reader’. RFID uses electromagnetic radiation to communicate between the transponder, commonly known as a ‘tag’, and the reader. Some RFID protocols allow the reader to both read and write data embedded on the tag; other RFID protocols are read-only. All data within the RFID system is carried on the tag [36].

RFID transponders are divided into two broad categories, active, and passive. Active transponders are powered with an on-board battery or other power source, while passive transponders are power-scavenging, drawing power from the electromagnetic radiation emitted by the reader. Although active transponders have a much larger range and data-transfer rate than passive transponders, passive transponders are much cheaper. This additional range is due to the full radio transceivers found in active transponders.

A third category, assisted passive, uses a battery when in range of the reader, and is passive otherwise.

RFID is commonly used in a wide variety of applications, including smart cards, livestock and pet tracking, asset control, and passports.

Unlike some wireless protocols, such as microwave transmissions, RFID is not always limited to line-of-sight communication between the tag and the reader. However, depending on the frequency used, transponders have a limited range of communication.

RFID transponders can be packaged in numerous sizes and shapes, as shown in figures 3.5 and 3.6.

3.10.2 RFID Reader to Transponder communication

When the RFID transponder is within range of an RFID reader, the activation signal broadcast by the reader activates the transponder. Passive RFID transponders then use the power gleaned from the reader’s signal to transmit any data on the microchip back to the reader. In this way, the reader and the transponder communicate.
Figure 3.5: An RFID transponder. [4]

Figure 3.6: RFID transponders of the type commonly used in libraries. [5]
3.11 Classification of Input Controls

While section 3.3 explored the use of sensors with TIO, the state information most commonly useful to a designer while prototyping control panels are common Human-Computer Interface controls. In this section I examine the types of input controls that will be used in the RFID IntelliTIO.

I classify Human Computer Interface input controls into two categories, *switched* and *valuated*. I make this distinction based on the state-awareness of the input control, not on the form factor; it is entirely possible for an input control from both categories to have an identical form function. Each of these categories can be further divided into subordinate classes.

3.11.1 Valuated Input Controls

Valuated input controls can either be *continuous* or *terminal*. Terminal valuated input controls have a value between two states (‘high’ and ‘low’), allowing the input control to exist in any value between the two states. This contrasts with continuous valuated input controls, such as ball-mice, in which the input control may be indefinitely increased in a direction.

Common form factors for valuated input controls are rotary dials or bidirectional sliders. Some common valuated input control examples include volume controls and joysticks. Terminal valuated input controls are much more common than continuous input controls.

Other examples of valuated input controls include:

- Air-conditioning dials
- Dimmer-switches for lighting control
- Tuning pegs for musical instruments

3.11.2 Switched Input Controls

In contrast to valuated input controls, which may exist in an infinite number of states generally between ‘high’ and ‘low’, the state of a *switched* input control exists as one of a finite set of discrete values — for example, either ‘on’ or ‘off’, or either ‘1’, ‘2’ or ‘3’. Switched Input Controls can be divided into two subcategories:

- Momentarily switched
- State-full switched

*Momentarily* switched input controls generally consist of an unstable and a stable state. When the input control is stimulated, it transitions from the stable to the unstable state and, when the stimulus is no longer present, or after a time period, it drops from the unstable state to the stable state. A common momentarily-switched input control
in HCI is a mouse button; the button consists of two states, ‘Clicked’ and ‘Not-clicked’. The default (stable) state of the input control is ‘not-clicked’; the user depresses the input control to transition to the unstable (‘clicked’) state and, when the stimulus is no longer present, the input control will return to the stable state. Common examples of \textit{momentarily} switched input controls in Human-Computer Interfaces include:

- Keyboard buttons
- Mouse buttons

However, when prototyping control panels, a much wider range of input controls must be considered. For example, momentarily switched input controls from common domestic life may include:

- Vehicle horns
- Doorbells

\textit{State-full} switched input controls consist of two or more stable states. Transitioning from one state to another requires external stimulus; however, unlike momentarily switched input controls, state-full switched input controls will stay in a state until acted upon by a stimulus. A common example of a state-full switched input control is a light switch; the switch consist of two states, ‘On’ and ‘Off’. A user can toggle the switch between these two states, but, once in a state, the switch will remain in that state indefinitely. Other examples of common state-full switched input controls include:

- On/Off controls. Examples include: light switches, stereo systems and cellular phones.
- Vehicle controls. Examples include: indicator controls, headlight controls.

A common form factor for state-full switched input controls are discretely-positioned rotary dials or bidirectional sliders, such as those commonly found on washing machine control panels, or electric stove-tops.

\textbf{3.11.3 Providing designers with Input Controls}

When designing prototypes, it is essential that the designer is provided the ability to integrate lifelike objects, functionality, haptic control and tactile feedback. In doing so, the designer is enabled to create realistic prototypes. Three things are essential, then, for the designer to make full use of the SAR prototyping work flow:

1. Form
2. Physical functionality
3. System functionality
In many situations, correct ergonomic form factor of the input controls is essential to the final design of the control interface. One such example is that of submarine control interfaces. The presence of numerous controls in a small control interface area creates challenges for the designer, particularly in the areas of usability and ergonomics. When prototyping such a control interface using Spatial Augmented Reality, having physical artifacts to position and arrange on the control surface allows the designer to determine problematic layouts caused by the shape and size of the control inputs. For example, the relative spatial positioning of a joystick and RADAR screen may result in occlusions from the users view-point; such a problem may be difficult to detect without the aid of a physical artifact to provide depth cues.

While form provides the designer with tactile cues, physical functionality greatly enhances the haptic feedback of the prototype. It is not enough for an input control to merely look like it’s real-world counterpart; the input control must also physically function as it’s real world counterpart. For example, users should be able to rotate rotary dials and slide bidirectional sliders. In addition, the haptic feedback of the control should mimic that of the fully-functional real-world counterpart; discretely-positioned rotary dials and two-state switches should ‘click’ into position.

The final essential characteristic of the input control is the system functionality. As much as possible, the SAR system should emulate the functionality of the input control. For example, if a prototype integrates a light-switch, the SAR system should emulate the lighting effect when the switch is ‘on’.

This chapter focuses on the final characteristic, system functionality.

3.11.4 System functionality

While some aspects of input control functionality can be emulated with user-tracking and other techniques, the additional complexity put on the SAR system inhibits fully-featured emulation of functionality. In addition, emulating hardware functionality in SAR software means an additional layer of work the designer must implement.

A better solution for this challenge, then, is the integration of functionality into hardware directly, while still maintaining several important characteristics of SAR Tangible Input Objects:

- TIO must be spatially dynamic
- TIO must be inexpensive to produce
- TIO must require minimum maintenance

Spatially dynamic

In the context of this thesis, the term “spatially dynamic” refers to the ability of the designer to move the TIO with relation to the substrate within the SAR scene, without the need for hardware reconfiguration. This contrasts with, for example, electronic hardware built directly on to the substrate, which can not easily be moved due to wiring.
and physical attachment to the substrate. Spatially dynamic TIO provide the designer with flexibility and the ability to rapidly try numerous TIO layouts, while maintaining functionality — aspects which are critical to the rapid prototyping work flow, but which are difficult to attain with traditional electronics.

Creating state-aware TIO through the integration of Radio-Frequency IDentification can meet these requirements, creating functional TIO that can be integrated into a SAR scene easily and dynamically.

Two input controls will be created, demonstrating both switched and valuated state management.
3.12 Choosing input controls for RFID integration

The integration of RFID with input controls allows the creation of input controls which are power-scavenging and state-aware. This is ideal for use within SAR systems.

While the number of differing form factors for input controls is almost infinite, the proof-of-concept creation of both a switched and valuated input control demonstrates that the integration of RFID with input controls is beneficial for rapid prototyping with SAR work flows; the form function of the input control can be trivially changed while the underlying functionality of the input control remains the same.

3.12.1 Two-state switched input devices

Creating a two-state switched input device using RFID can be achieved by simply turning the RFID transponder off or on, thus adding or removing the signal from the scene. The SAR system is then able to interpret the lack of signal as an ‘OFF’ state, and interpret the transponder signal presence as an ‘ON’ state.

Turning the transponder signal on or off can be achieved by simply bridging over the antenna, short-circuiting it. This has the effect of turning off the transponders signal. A two-state switch can thus be created, with the ‘ON’ state of the switch corresponding to the lack of bridging over the transponder, and the ‘OFF’ state corresponding to the transponders antenna being short circuited.

3.12.2 Valuated input devices

The creation of valuated input devices allows the integration of a wide range of input devices into the SAR scene. Input devices such as joysticks, radial dials, valuated sliders and multi-selection switches can all be created from a valuated input device; indeed, the aforementioned devices can be conceptually considered differing form-factors of the same input device. By combining an astable multivibrator with an RFID transponder in the TIO, the analog state position of the valuated TIO can be transmitted to the SAR system. A multivibrator consists of several amplifying components, such as transistors, cross-coupled with several resistors. Several different types of multivibrators exist, including:

**Astable:** two unstable states. States are constantly oscillated between without requiring external input.

**Bistable:** two stable states. External stimulus is required to switch state.

**Monostable:** one stable, one unstable state. External stimulus is required to switch into the unstable state; however, the multivibrator will switch to the stable state after a period of time without external stimulus.

By integrating a potentiometer into the astable multivibrator, the time spent in the unstable state can be changed, allowing the state of the variable resistor to be determined. This can be used to switch the RFID transponder signal off and on, allowing
Figure 3.7: Schematic of the Astable Multivibrator [6]. Note that $R_2$ is replaced by a potentiometer.

the SAR system to determine the valuated input state by measuring the time between the transponder signal change of state.

Construction of the astable multivibrator

The astable multivibrator is constructed from the schematic shown in figure 3.7.

Time spent in the unstable state

The state of the input control is determined by the value of the potentiometer. To communicate this value with the SAR system, the input control varies the time spent in the unstable state, causing the RFID transponder to disappear and reappear in the RFID readers vision as the multivibrator enables and disables the RFID transponders antenna.

The minimum and maximum time spent in the unstable state can be calculated using the maximum and minimum resistance of the potentiometer. The time spent in each state of the multivibrator is given by:

$$T_1 = \log 2R_2C_1$$
$$T_2 = \log 2R_3C_2.$$

As the potentiometer replaces $R_2$ in the above schematic, the maximum and minimum time periods spent in the unstable state are given by $T_{\text{max}} = \log 2R_{2\text{max}}C_1$ and $T_{\text{min}} = \log 2R_{2\text{min}}C_1$ respectively, where $R_{2\text{max}}$ and $R_{2\text{min}}$ refer to the maximum and
minimum resistances of the potentiometer.
3.13 RFID Antenna Design

The range of possible RFID reader antenna designs is quite varied, ranging from finger-mounted antenna as used by Thomas et al. [7] to large, mounted antenna. However, in contrast to Thomas et al. [7], I demonstrate a mounted antenna, rather than extending on the finger-mounted antenna. This provides several benefits over the system demonstrated in [7]:

- Constant power to devices within the antennas field, and
- Collaborative environment benefits.

The demonstrated system is placed beneath the SAR substrate, with the IntelliTIO placed above the substrate.

The type of RFID system in use is important with regards to the design of the antenna. Radiative RFID systems use an antenna of comparable size to the wavelength, and utilize a much greater range between the antenna and transponder. Inductive RFID systems, on the other hand, use a much smaller antenna compared to the wavelength, and have a smaller maximum range between antenna and transponder.

The work presented in this thesis makes use of an inductive RFID system.

3.13.1 Antenna Design

While there are a variety of RFID reader antenna designs, I demonstrate the valuated and switched input controls with a single-loop antenna, depicted in figure 3.8. For optimum performance, the antenna should have an impedance of 50 Ohms [37]. To accomplish this, I utilize a gamma-matched network [37]. The combination of single-loop gamma-matched network was chosen over a T-matching network for its inexpensive and simplistic nature [37]. In order to gain maximum performance from the antenna, some tuning is required, by properly impedance matching the antenna to the RFID reader. A base-resistance capacitor, denoted the ‘swamping’ capacitor, was inserted into the loop, along with two variable-capacitors. These variable-capacitors are used along with the gamma-matched network to fine-tune the impedance of the antenna. The tuning capacitors are depicted in figure 3.9 when attached to an antenna analyzer.

Fine-tuning antenna impedance

The impedance of the antenna is tuned in two ways. First, the gamma-matched network ‘matching tap’ is connected to the antenna. The position of the matching tap depends on the impedance of the antenna. Secondly, the variable-resistance capacitors are manipulated to lower the quality factor, denoted $Q$ of the antenna. To achieve this, the antenna is attached to an antenna analyzer, shown in in 3.10. The antenna analyzer model shown is a MFJ-269. The $Q$ should be 50 for optimum performance [37].
Figure 3.8: Single-loop RFID antenna with a gamma-matched network

Figure 3.9: Two variable capacitors and the swamping resistor
3.14 Input control functionality over RFID

The valuated and switched input controls outlined in section 3.12.1 and 3.12.2 were attached to an RFID transponder, shown in figure 3.11. This transponder was chosen for its size, which allows relatively easy access to the antenna. Access points were attached at each end of the transponders antenna such that, when the access points were connected, the antenna loop was shorted, as seen in 3.12.

The two input controls were then attached to the access-points.

3.14.1 Switched input control

The switched input control utilizes a switched gate, toggled by the user. When the gate is closed, the two access-points are connected, shorting the transponders internal antenna. This in turn prevents the transponder from returning the signal received by the reader, removing the transponder from the readers field of view. The SAR system determines the state of the input control by checking whether the input control is visible from the RFID reader. A visible input control indicates an ‘on’ state, whilst a ‘missing’ transponder indicates an ‘off’ state.
Figure 3.11: The RFID transponder

Figure 3.12: Access points attached to the transponder's internal antenna
3.14.2 Valuated input control

The valuated input control utilizes a astable multivibrator which outputs a square-wave. The wave controls the switch, oscillating the IntelliTIO within the readers field of view. The time between each oscillation is controlled through the variable resistance supplied in this case by a rotary-dial potentiometer. This can be seen in figure 3.13 and 3.14. By measuring the time between each oscillation, the SAR system is able to determine the value of the input control.

3.14.3 Generalizing for input controls

Additional input controls can be utilized with either of the two methods of RFID signal manipulation demonstrated here. The valuated input control method of timed signal oscillation, for example, is applicable not only to rotary-dial valuators, but also for multi-state switches and sliders. To fully implement a multi-state switch using the methodology presented here, the oscillation time at each switch-state can be calculated using the formula presented in 3.12.2 allowing the SAR system to accurately determine the switch state.
Figure 3.14: Relatively low frequency square-wave, generated by the astable-multivibrator. Note the change in frequency from 3.13 is determined by the changed resistance of the potentiometer.
3.15 Analysis

The RFID technology presented in this chapter has several benefits when compared to that used by Thomas et al. [7]:

3.15.1 Benefits

- Improved collaborative model,
- Simpler to integrate into the SAR system,
- Vastly improved haptic and tactile feedback, and
- Allows for more complex TIO use

Improved collaborative model

The system presented in [7] utilizes a finger-mounted antenna. Whilst this provides some benefits, it inherently limits the collaborative aspect of the SAR prototyping system. As the finger-mounted antenna is crucial for the use of the system, the system is limited to a single user, although additional users could utilize multiple gloves. By contrast, the system presented here does not depend on a user-specific hardware, and thus allows multiple users to use the system concurrently.

Simpler integration into the SAR system

The system presented in [7] emulates TIO functionality through finger-mounted RFID antenna and RFID transponder proximity. To ‘press’ a button, the user brings the finger-mounted antenna into the proximity of the RFID tag. This relies on software emulation to provide TIO functionality. By contrast, the IntelliTIO presented here can be viewed by the SAR system as ‘black boxes’, informing the SAR system of their state, and offloading the input control functionality to the tangible input objects.

Improved haptic and tactile feedback

The IntelliTIO presented here utilize ‘real-world’ input controls, such as switches and rotary dials, as their form-factor. This provides the user with identical haptic and tactile feedback when using the prototype as she would experience with the final, developed product. This improves on the emulated input controls used in [7] and vastly increases the immersion and realism of the SAR prototype.

Allows for more complex TIO use

The TIO used in [7] are simplistic, and do not provide functionality beyond emulated state-switching. The IntelliTIO presented here provide full switched and valuated functionality.
3.15.2 Drawbacks

When compared to the IntelliTIO presented in 3.3, the technology presented in this chapter has a drawback which must be taken into consideration when designing IntelliTIO; namely, the requirement that the IntelliTIO have a low power draw. This limits the number and complexity of the sensors which can be placed on the TIO, which may limit their usefulness in some prototyping applications.

The RFID technology may present some additional drawbacks, including data rate or latency, particularly when utilizing multiple TIO within a single SAR scene. The magnitude and effects of these potential drawbacks should be investigated in future work in this area. The limited range and bandwidth of the TIO could perhaps also be considered drawbacks.
3.16 Conclusions

This chapter has demonstrated that power-scavenging IntelliTIO can be created, providing ‘traditional’ input control functionality to the SAR prototyping realm. The IntelliTIO demonstrated here are inexpensive, and can be created with a vast number of form factors, making them suitable for use in a wide range of prototyping scenarios.
Chapter 4

Summary Conclusions
4.1 Conclusions

In the realm of spatially augmented reality (SAR) assisted rapid prototyping, the use of tangible input objects (TIO) provides the designer with the ability to create a realistic, immersive SAR scene. As the interaction between the designer and the prototype occurs through the TIO, it is imperative that the TIO utilized in prototypes have a high degree of fidelity and functionality. This is particularly important prototyping input-centric objects, such as control panels, where the usability of the prototype is heavily dependant on the tactile and haptic feedback of the input controls. Traditional SAR input controls feature ‘dumb’ input controls with limited and emulated functionality. These types of input controls do not provide the full functionality of their real-world counterparts, and, in addition, when functionality must be emulated by the SAR system, these input controls impose additional constraints on the prototyping work flow.

Transitioning from a ‘dumb’ tangible input object to an object aware of both self-state and the local SAR environment can greatly enhance the SAR scene, providing a level of fidelity and functionality to the prototype unattainable through the use of ‘dumb’ TIO alone.

When creating an intelligent tangible input object (IntelliTIO), the choice of sensor is crucial in providing the foundation for the design of the IntelliTIO. The choice of sensor dictates the level of intelligence of the TIO, the types of state that the TIO is aware of, and imposes limits on the implementation of the IntelliTIO.

By integrating sensors with the tangible input object, the TIO can be provided with awareness of the local SAR environment, including such information as spatial position, temperature, proximity with other TIO and local environmental lighting. Integrating such sensors with TIO imposes some challenges, however, particularly with regards to the power draw of the IntelliTIO.

In chapter I, I explore the implementation of a TIO with local environmental awareness, utilizing an open, extensible hardware platform which allows future work to easily build on my results. The created IntelliTIO, deemed a ‘Blob’, demonstrates some possible capabilities of sensor-integrated TIO as well as exploring visible light as an IntelliTIO to SAR system communications medium. Whilst the Blob implementation has several drawbacks, particularly with regards to the limited functionality of the sensors utilized, it does competently demonstrate that IntelliTIO have a viable place in SAR infrastructure.

In addition to local environmentally-aware TIO, an important consideration for SAR input objects is the modelling of real-world functionality. For input controls, this is particularly important, as the input controls are largely responsible for providing the bridge between the user and the SAR system. However, providing real-world functionality in SAR systems is difficult through emulation alone, and ‘dumb’ input controls do not exhibit fully-functional behaviour. To overcome this, I categorize input controls into switched and valuated state, and explore the integration of both switched and valuated states with TIO. In addition, I examine the use of RFID as both a communications medium and a power source for the resulting IntelliTIO, improving on existing TIO and demonstrating viable, functional input controls for SAR prototyping.
This thesis has demonstrated that functionality can be added to Tangible Input Objects to make them more useful within the realm of SAR-based rapid prototyping. The method of integrating sensors into the TIO, demonstrated with the Blob, provides benefits to both SAR-based prototyping work flows and SAR systems, and is a viable, usable technology. Other hardware-based frameworks for creating sensor-based objects, such as Arduino [38], could also be used, allowing for designers without a deep background in electrical engineering to quickly and cheaply create functional input objects and prototypes with integrated sensors. While the demonstrated Blob has several drawbacks, these can be overcome with more resilient hardware — particularly better visible-light filters — and the advantages provided to the SAR system are numerous. Future work could build on either the Wasa Board or other, similar technologies, to create a framework of SAR-based sensors and software, allowing developers to ‘mix and match’ their desired functionality, both in hardware and software. Such a framework could also take into account the use of multiple Blobs within a single SAR scene, as the use of, and interactions between, more than one Blob would be invaluable to the SAR-based prototyping work flow.

The RFID technologies demonstrated in this thesis build on previous work, providing several immediate benefits, such as multi-user collaboration without the need for user-specific hardware as was used in [7, 32]. Although the RFID technologies do require slightly more electrical engineering knowledge than those of the Blob, the ability to provide functional sliders and rotary dials is both novel and invaluable to the SAR field. Within this novel application, there are many avenues for future work. For example:

- Mass-producing multivibrators with integrated potentiometers should be investigated.
- The use of different antenna designs, with emphasis on re-usability and non-planar application, is a key area of interest. Non-planar substrates are often used in prototypes — car dashboards, for example, are almost entirely curved — and such substrates may present challenges with ‘typical’ table-top antenna mounted designs.
- Optimization of the RFID frequency, antenna range and TIO bandwidth could be investigated, especially when considering multiple RFID-enabled TIO within a SAR scene.

The use of RFID within SAR to supply Tangible Input Object functionality has immediate applications, and is novel and exciting. The Intelligent Tangible Input Objects presented here bridge the gap between SAR prototypes and their functional, real-world counterparts. They improve on existing ‘dumb’ interface objects and utilize novel techniques to provide state-awareness within SAR environments. The IntelliTIO presented here help to bridge the gap between the virtual and physical worlds, creating a better interface for human-computer interaction, and drawing us ever closer to the seamless interaction of the virtual utopia presented by Sutherland in 1965.
Appendix A

Wasa Board AT Commands
<table>
<thead>
<tr>
<th>Command</th>
<th>AT Command</th>
<th>User values</th>
<th>Default</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Echo Commands</td>
<td>Q</td>
<td>“0,1”</td>
<td>1</td>
<td>E1 will echo all characters received from a connected host E0 will turn off character echo</td>
</tr>
<tr>
<td>Quiet Result Response</td>
<td>E</td>
<td>“0,1”</td>
<td>0</td>
<td>Q1 will turn off result code response Q0 will allow result codes to be sent</td>
</tr>
<tr>
<td>Verbose Result Response</td>
<td>V</td>
<td>“0,1”</td>
<td>1</td>
<td>V1 will enable verbose result codes V0 will enable numeric result codes. Currently supported result codes are: OK (verbose) 0 (numeric) ERROR (verbose) 4 (numeric) EVENT (verbose) 7 (numeric) DATA (verbose) 8 (numeric)</td>
</tr>
</tbody>
</table>

Table A.1: Basic Wasa Board AT commands
<table>
<thead>
<tr>
<th>S-Register Commands</th>
<th>Command (Data in or out)</th>
<th>Command (Direction and set up)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital GPIO0</td>
<td>S130</td>
<td>S120</td>
<td>For all Digital GPIOs: S12x = 0 means GPIO is an input, no pull up or down</td>
</tr>
<tr>
<td>Digital GPIO1</td>
<td>S131</td>
<td>S121</td>
<td>S12x = 1 means GPIO is an input with internal pull up</td>
</tr>
<tr>
<td>Digital GPIO2</td>
<td>S132</td>
<td>S122</td>
<td>S12x = 2 means GPIO is an input with internal pull down</td>
</tr>
<tr>
<td>Digital GPIO3</td>
<td>S133</td>
<td>S123</td>
<td>S12x = 3 means GPIO is an output</td>
</tr>
<tr>
<td>Digital GPIO4</td>
<td>S134</td>
<td>S124</td>
<td></td>
</tr>
<tr>
<td>Digital GPIO5</td>
<td>S135</td>
<td>S125</td>
<td></td>
</tr>
<tr>
<td>Digital GPIO6</td>
<td>S136</td>
<td>S126</td>
<td></td>
</tr>
<tr>
<td>Digital GPIO7</td>
<td>S137</td>
<td>S127</td>
<td></td>
</tr>
<tr>
<td>Analog AN0</td>
<td>S200</td>
<td>Read only</td>
<td>This is also the built in light sensor</td>
</tr>
<tr>
<td>Analog AN1</td>
<td>S201</td>
<td>Read only</td>
<td></td>
</tr>
<tr>
<td>Analog AN2</td>
<td>S202</td>
<td>Read only</td>
<td></td>
</tr>
<tr>
<td>Analog AN3</td>
<td>S203</td>
<td>Read only</td>
<td></td>
</tr>
<tr>
<td>Analog AN4</td>
<td>S204</td>
<td>Read only</td>
<td></td>
</tr>
<tr>
<td>Analog AN5</td>
<td>S205</td>
<td>Read only</td>
<td></td>
</tr>
<tr>
<td>Analog AN6</td>
<td>S206</td>
<td>Read only</td>
<td></td>
</tr>
<tr>
<td>Analog AN7</td>
<td>S207</td>
<td>Read only</td>
<td></td>
</tr>
</tbody>
</table>

Table A.2: Wasa Board AT commands for reading and writing to the digital and analogue pins
<table>
<thead>
<tr>
<th>Extended Commands</th>
<th>Command</th>
<th>User set values</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wasa Board info</td>
<td>+GMM</td>
<td>Read Only</td>
<td>Requests Wasa board hardware model version</td>
</tr>
<tr>
<td>Wasa Board info</td>
<td>+GMR</td>
<td>Read Only</td>
<td>Requests Wasa board AT command software revision</td>
</tr>
<tr>
<td>Accelerometer</td>
<td>+OAA</td>
<td>1=active (default)</td>
<td>Places accelerometer into active mode or low power standby</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0=standby</td>
<td></td>
</tr>
<tr>
<td>Accelerometer</td>
<td>+OAW</td>
<td>Read Only</td>
<td>Requests accelerometer values for X, Y and Z</td>
</tr>
<tr>
<td>Data streaming, rate</td>
<td>+OSR</td>
<td>Multiple</td>
<td>This sets the data streaming rate.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Allowable values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0 = Streaming off</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 = 1 Hz, 2 = 2 Hz, 4 = 4 Hz,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>8 = 8 Hz, 16 = 16 Hz 32 = 32 Hz,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>64 = 64 Hz, 120 = 120 Hz</td>
</tr>
<tr>
<td>Data streaming, acceleration</td>
<td>+OSX</td>
<td>Unsigned 8-bit value.</td>
<td>Bit mask which indicates the accelerometer data channels to be streamed out.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Bit 0 : X acceleration data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Bit 1 : Y acceleration data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Bit 2 : Z acceleration data</td>
</tr>
<tr>
<td>Data Streaming, analogue values</td>
<td>+OSA</td>
<td>Unsigned 8-bit value.</td>
<td>Bit mask which indicates analogue data channels to be streamed out.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Bit 0 = analogue channel 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Bit 1 = analogue channel 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>etc.</td>
</tr>
<tr>
<td>Data streaming, ditigal values</td>
<td>+OSD</td>
<td>Unsigned 8-bit value</td>
<td>Bit mask which indicates digital data channels to be streamed out. If enabled all GPIO bits are streamed as a single 8-bit value, where GPIO0 is the LSB and GPIO7 is the MSB.</td>
</tr>
</tbody>
</table>

Table A.3: Extended Wasa Board AT commands available
Appendix B

Wasa Board Control Loop

```plaintext
function run:
  init() {
    setflags {
      write("ate0") // Turn off echo response
      write("atv1") // Turn on verbose response
    }
  }

  // Sample and set thresholds
  for i in 15 do:
    sampleAcceleration()
    sampleLDR()
  done

  averageAcceleration()
  averageLDR()

  // Enter the main loop
  while (true) do:
    tickstring = fetchTickString()
    parseString(tickString)

    checkForAcceleration(bool acceleration_occuring)
    checkForLDR1On(bool LDR1_on)
    checkForLDR2On(bool LDR2_on)

    toggleLED (1, acceleration_occuring )
    if (!LDR1_on && LDR2_on) // we’re under filter-colored light
      toggleLED (2, LDR2_on )
    else
      toggleLED (2, false ) // Toggle it off if we’re not
      // under filter-colored light.
    done
  end function run
```
Bibliography


[38] Getting Started with Arduino.