**Design Portfolio**

**By:**

**Net 1 Bier**

**Randolph Bock 25905570**

**FJ Fourie 26047799**

**Anton Durandt 25645013**

Submitted in pursuit of the degree

**BACHELOR OF ENGINEERING**

**In**

**COMPUTER AND ELECTRONIC ENGINEERING**

**North-West -University Potchefstroom Campus**

Supervisor: Prof J. Holm

Potchefstroom

October 23, 2017

**Contents**

[1 System Operational Requirements 1](#_Toc496518638)

[1.1 Functional Analysis – Operational Level Architecture and Behaviour 1](#_Toc496518639)

[1.2 Physical Requirements (Form): 4](#_Toc496518640)

[1.3 Interface requirements(Fit): 4](#_Toc496518641)

[1.4 Additional Requirements: 5](#_Toc496518642)

[2 Excel Project management 6](#_Toc496518643)

[2.1 Combined Project Documents 6](#_Toc496518644)

[2.1.1 Work Breakdown Allocation 7](#_Toc496518645)

[2.1.2 Project Schedule 8](#_Toc496518646)

[2.1.3 Progress tracking 8](#_Toc496518647)

[2.2 Risk / Mitigation Register 10](#_Toc496518648)

[2.3 Minutes of Meetings 11](#_Toc496518649)

[2.4 Randolph Bock 16](#_Toc496518650)

[2.4.1 Excel Project Management Documents 16](#_Toc496518651)

[2.4.2 Experience Report 19](#_Toc496518652)

[2.5 Anton Durandt 19](#_Toc496518653)

[2.6 FJ Fourie 20](#_Toc496518654)

[2.6.1 Excel Project Management Documents 20](#_Toc496518655)

[2.6.2 Experience Report 22](#_Toc496518656)

[3 Engineering Methods/Skills/Tools: 23](#_Toc496518657)

[3.1 Randolph Bock 23](#_Toc496518658)

[3.2 Anton Durandt 23](#_Toc496518659)

[3.3 FJ Fourie 23](#_Toc496518660)

[4 Sub-system specification DOCUMENTS: 23](#_Toc496518661)

[4.1 Randolph Bock 23](#_Toc496518662)

[4.2 Anton Durandt 24](#_Toc496518663)

[4.3 FJ Fourie 24](#_Toc496518664)

[4.3.1 Sub-system functional Analysis 24](#_Toc496518665)

[4.3.2 Sub-system Interface Definitions 26](#_Toc496518666)

[5 Design Documentation: 27](#_Toc496518667)

[5.1 System Design Documaentation 27](#_Toc496518668)

[5.1.1 Final System Functional Definition: 27](#_Toc496518669)

[5.1.2 System concept drawings: 27](#_Toc496518670)

[5.1.3 System interface Definitions: 28](#_Toc496518671)

[5.1.4 System integration testing: 30](#_Toc496518672)

[5.2 Sub-system Design Documentation Randolph Bock: 31](#_Toc496518673)

[5.2.1 Technology Survey/Datasheets 31](#_Toc496518674)

[5.2.2 Application Notes 41](#_Toc496518675)

[5.2.3 Trade off Studies 43](#_Toc496518676)

[5.2.4 Design Drawings: 44](#_Toc496518677)

[5.2.5 Behavioural Modelling: 45](#_Toc496518678)

[5.2.6 Design implementation: 47](#_Toc496518679)

[5.2.7 Sub-system Test and Evaluation: 48](#_Toc496518680)

**List of Figures:**

[Figure 1: System Operational Architecture 2](#_Toc496518681)

[Figure 2: System Operational Flow 3](#_Toc496518682)

[Figure 4: Interface of sub system 24](#_Toc496518683)

[Figure 5: Functional analysis of sub system 25](#_Toc496518684)

**List of Tables:**

[Table 1: Work Breakdown and Allocation 7](#_Toc496518685)

[Table 2: Project Schedule 8](#_Toc496518686)

[Table 3: Progress tracking 9](#_Toc496518687)

[Table 4: Risks and Mitigations 10](#_Toc496518688)

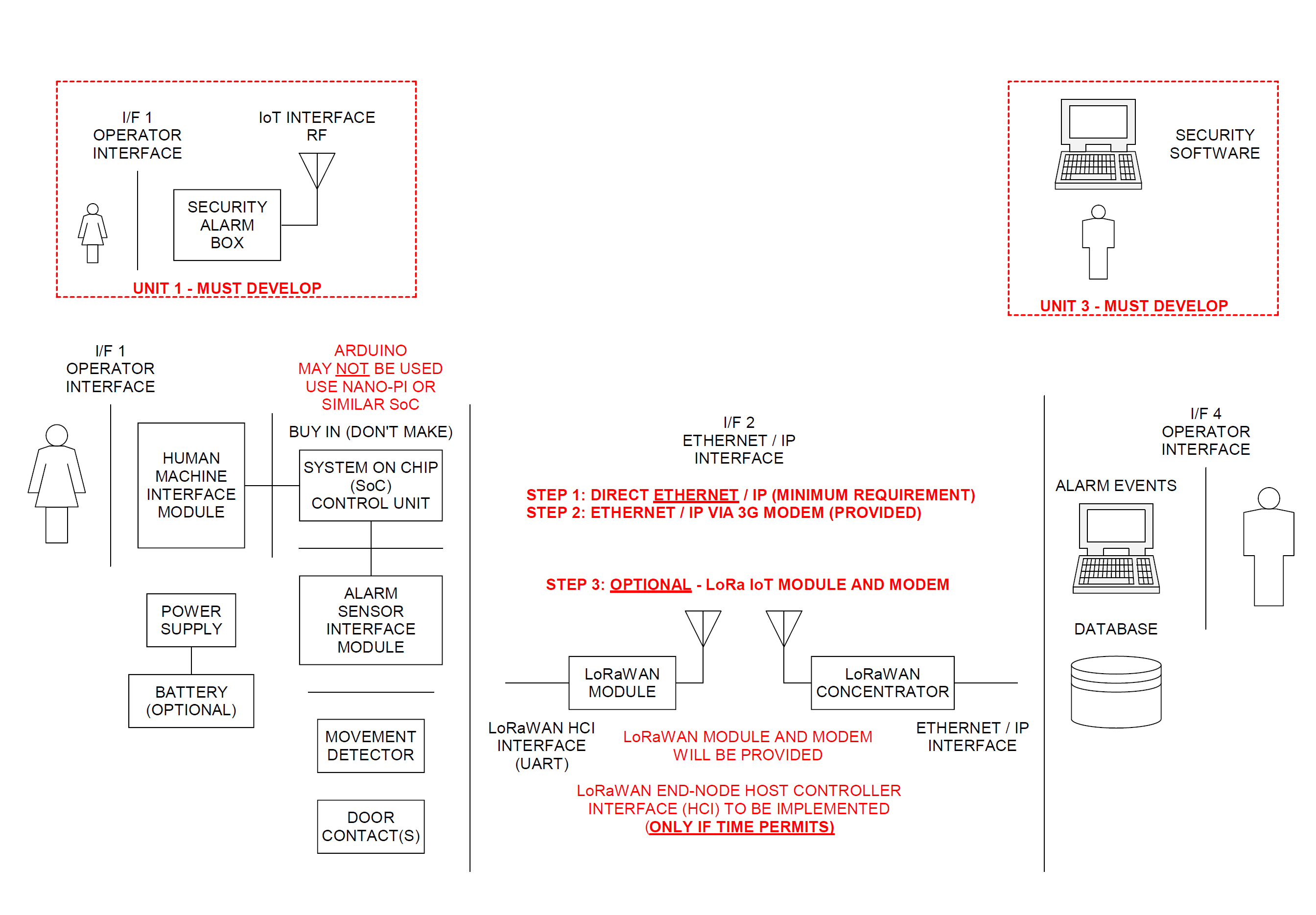
[Table 5: Timeline of Randolph Bock 18](#_Toc496518689)

[Table 6: Timeline of FJ Fourie 21](#_Toc496518690)

# 1 System Operational Requirements

## 1.1 Functional Analysis – Operational Level Architecture and Behaviour

1.1.1 System Operational Architecture:



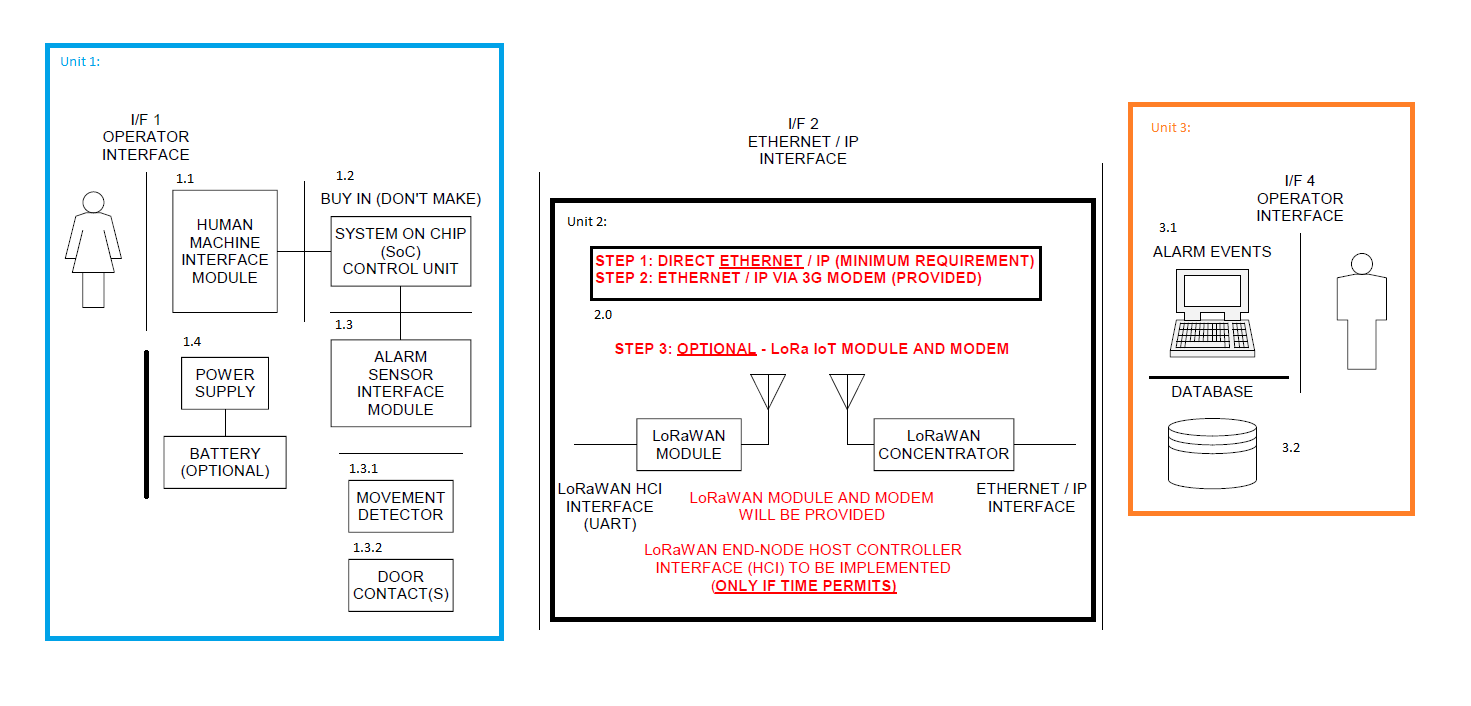


Figure : System Operational Architecture

1.1.2 System Operational Flow:

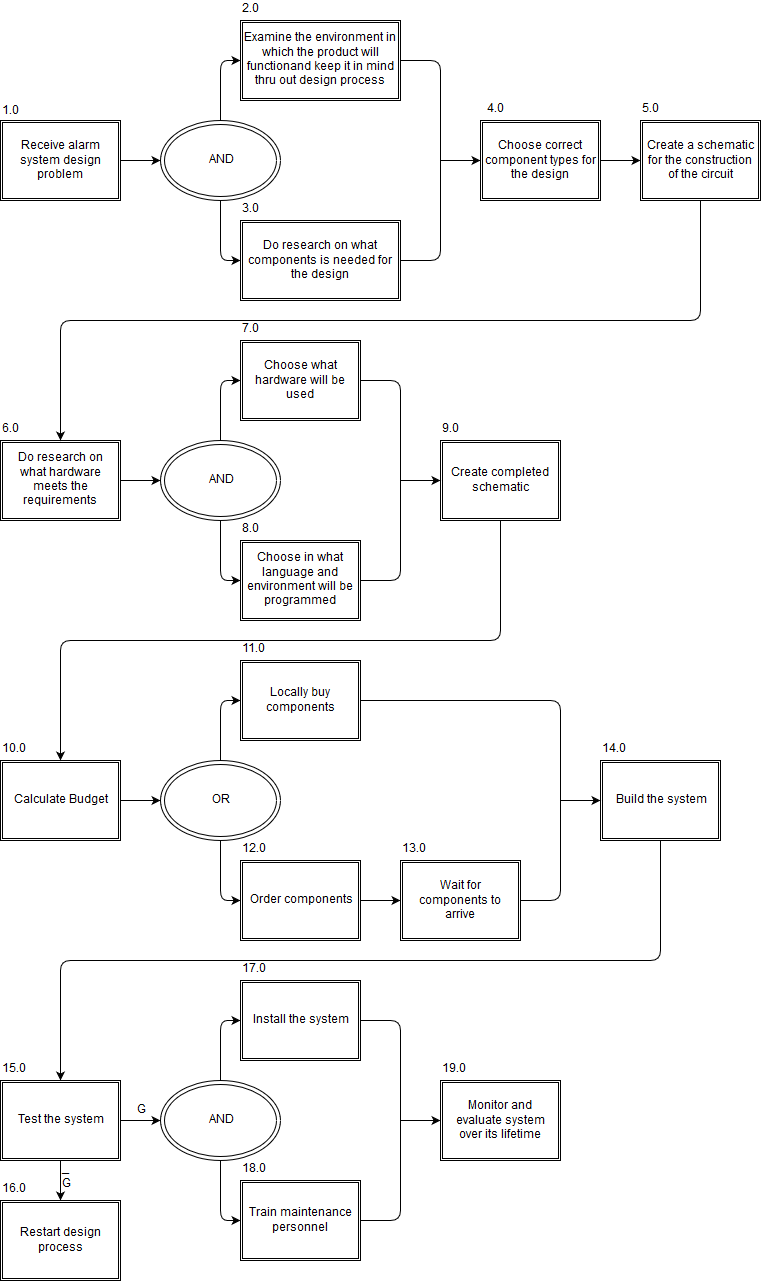


Figure : System Operational Flow

## 1.2 Physical Requirements (Form):

* The alarm panel as well as any sensors of the system must be IP54 rated.
* The alarm panel must be wall mountable
  + The Design must not be to heavy
* Unit should not protrude at strange angles that will cause harm to people passing by
* Functional shape that will fit into any home without inconvenience
* Size of unit must not be to large
  + 1m x 1m x 0,5m size constraint
* The alarm panel must allow for wires to enter the panel without damaging the wires

## 1.3 Interface requirements(Fit):

* The interface between the end user and the alarm will be both audio and visual, and be easily understandable
* All items on the functional architecture must be developed, apart from PIR and other sensors
* The single board computer (SBC) will be a Raspberry Pi
* The database server will initially be hosted on a laptop or student PC. Later implementation should be in the cloud, but cloud implementation is not a prerequisite to pass the module. However, clear proof must be provided that the database and monitoring software on the PC has been designed by following the design process
* An output will be provided to the Power Block to show that motion has been detected on an “outdoor sensor”
* The input power to the Alarm Panel will be provided from the Power Block and will be 12V at a maximum of 6W
* The Alarm Panel will provide its own internal voltages for the SBC and other components
* The IoT link will be a transparent GSM link that replaces the Ethernet. LoRaWAN can be used at the end of the project if time allows
* The interface between the incoming power and the internal power supply of the system will be a two-wire interface, positive and negative wire, to be connected to the system with screw in wire terminals.
* The placement on the wall cannot be too high as it will make it difficult for users to access the panels
* The placement cannot be to low either as it may become a risk for small children
* The placement should not be over pre-existing water pipes or wiring in the wall

## 1.4 Additional Requirements:

1.4.1 Environmental Requirements:

* The alarm panel as well as any sensors of the system must be IP54 rated
* Will be made reasonably tamper proof
* Will be made from strong material to endure some wear
* The system must be protected against Electro Static Discharges, ESD.

1.4.2 Safety requirements

* The system will be grounded to ensure no electrical shocks can occur to an end user
* No open wires
* Not accessible to children
* No places where people can get shocked
* Closed box so that people who do not understand the device cannot tamper with it
* Reasonably tamper proof

1.43 Legislative Requirements (SAIDSA bylaw 25)

* Control equipment
  + Control panel installed min of 1.5 m form ceiling
  + Digital keypads must be of the data transfer technology type
  + Disarming delay no more than 30 seconds
* Signalling equipment
  + Signalling equipment will be positioned within the protected area
  + Not placed where telephone lines are vulnerable
* Maintenance
  + Inspect and test each detection device back to control panel
  + Inspecting alarm panel and transmitter
  + Inspect cables for visible damage

1.4.4 Usability Requirements:

* System must be easily operated with minimal training required to operate the system
* Compensation for mounting will be made for the control box

# 2 Excel Project management

## 2.1 Combined Project Documents

The following documentation has been taken and adapted from the project management excel document.

### 2.1.1 Work Breakdown Allocation

In the following Table 1 the work allocation for each member is documented. This work allocation was done at the start of the project and as new work may arise or some work may prove to be more difficult than thought or some work may be found to be very easy this list will be adjusted accordingly.

Table : Work Breakdown and Allocation

|  |  |
| --- | --- |
| **Member** | **Work allocation** |
| Randolph Bock | * Design circuit for security inputs (Including Door switches and PIR's) * Design circuit for output siren (to be run on 12v) * Design and implement a user interface * Design and code the SoC to handle all inputs and outputs * Order parts * Trade-off decisions * Assemble the final unit 1 as a whole |
| Anton Durandt | * Design and implement a Direct Ethernet / IP connection between SoC and Pc * Design and implement an Ethernet / IP connection via 3g between SoC and Pc * Designing and implementing the database to the cloud * Designing and implementing the database communications |
| FJ Fourie | * Design database for information of users and alarm systems * Design backend operator interface * Write program for backend of system to manage alarms going off and instruct operators on what to do * Integrate backend program with database * Ensure operator interface works correctly with signals received from web * Assemble the final complete project |

### 2.1.2 Project Schedule

Below in Table 2 the timeline for the project is documented as decided upon at the start of the project.

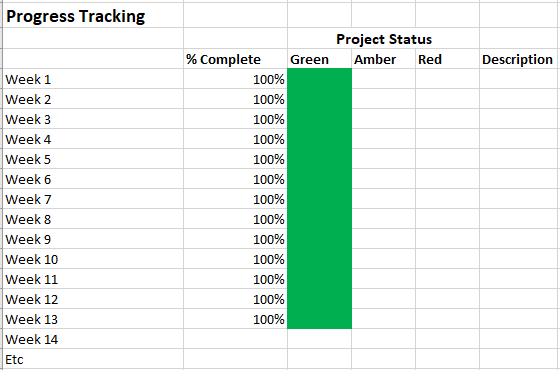
Table : Project Schedule

|  |  |
| --- | --- |
| **Date** | **Description** |
| **17-Jul-17** | **Project Initiation** |
| **10-Aug-17** | **Functional Analysis Test** |
| **14-Aug-17** | **Preliminary Design Complete** |
| **31-Aug-17** | **30% Completion Milestone** |
| **30-Sep-17** | **50% Completion Milestone** |
| **27-Oct-17** | **80% Completion Milestone** |
| **10-Nov-17** | **100% Completion Milestone** |
| **16-Nov-17 to 17-Nov-17** | **Demonstration** |

### 2.1.3 Progress tracking

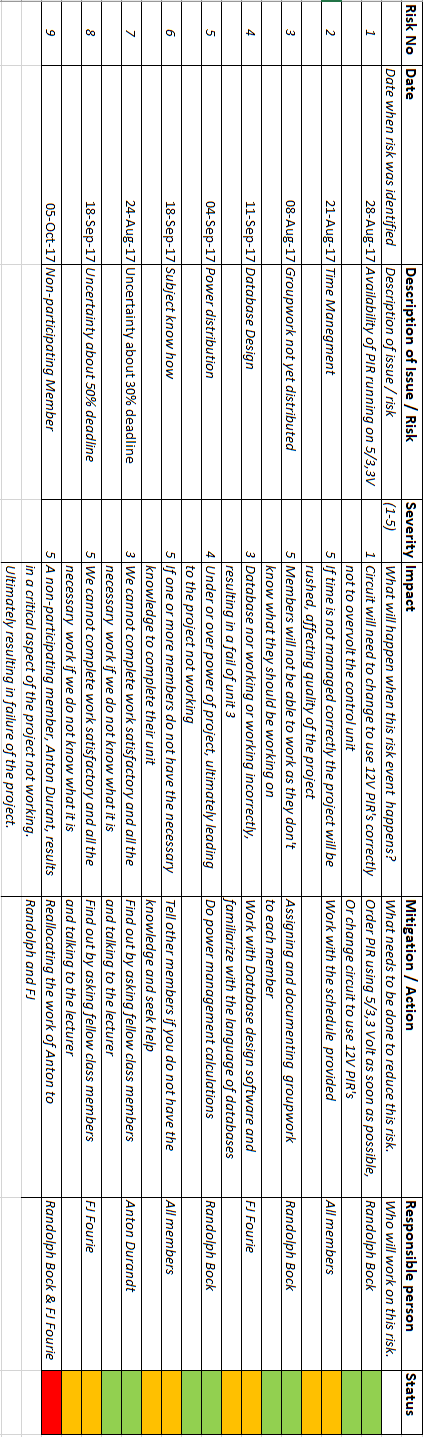
We use a RAG status to track the progress of the project each week with descriptions for when the project falls into the amber or red regions. This is all shown below in Table 3.

Table : Progress tracking



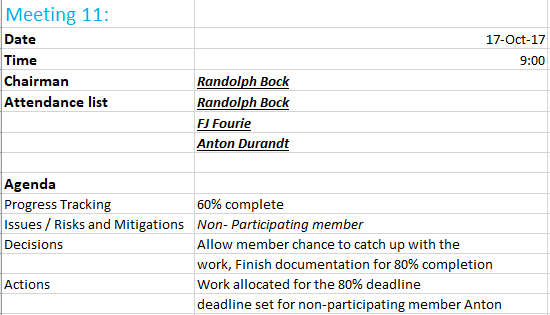
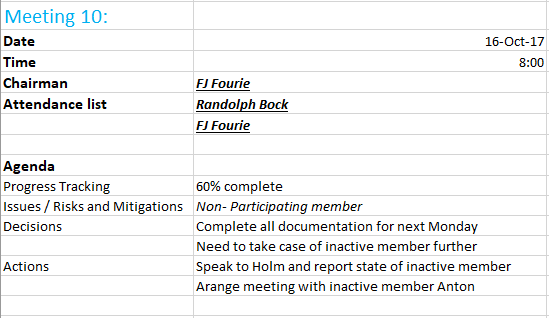
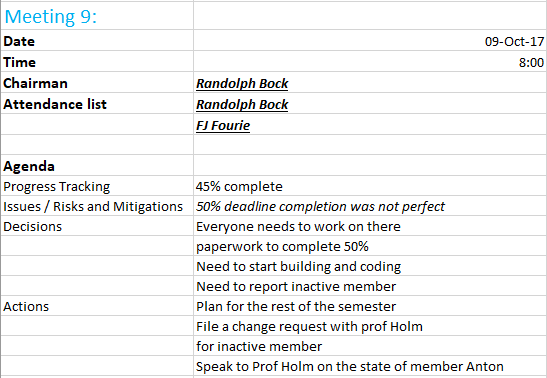
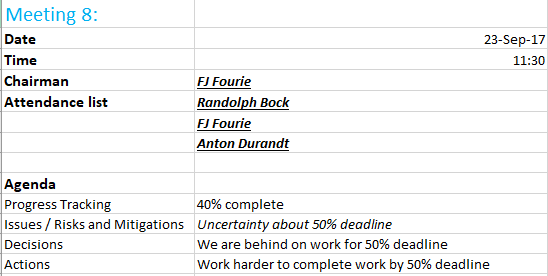
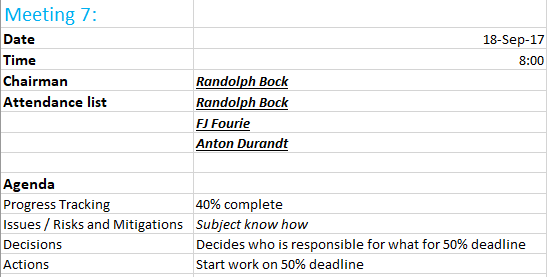
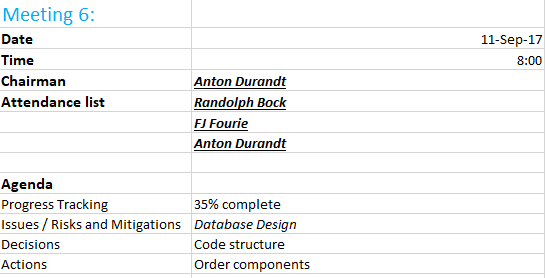
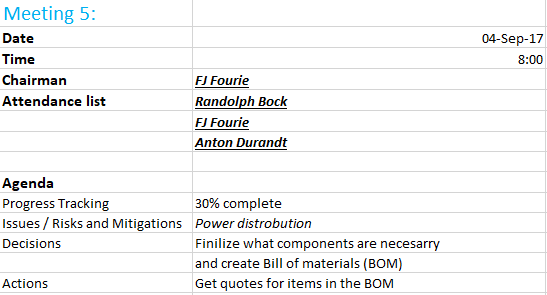
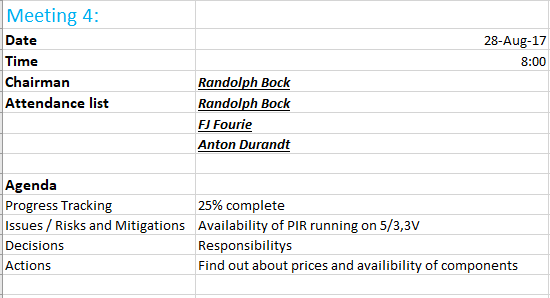
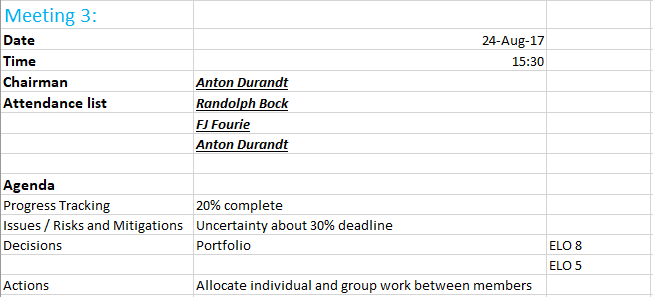
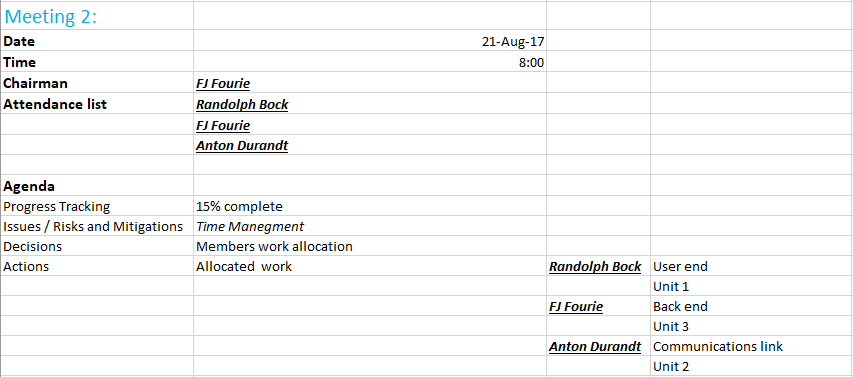
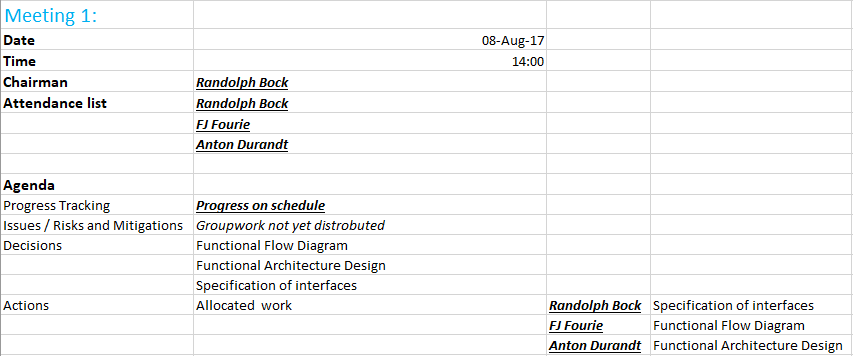
## 2.2 Risk / Mitigation Register

Table : Risks and Mitigations



## 2.3 Minutes of Meetings

The following are the minutes of all the meetings held by the group.



## 2.4 Randolph Bock

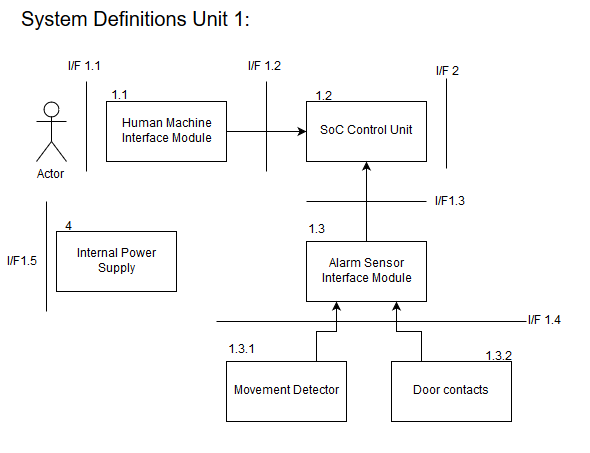
### 2.4.1 Excel Project Management Documents

#### 2.4.1.1 Expertise

* Software
* Hardware
* Documentation

#### 2.4.1.2 Parts of the project the member is responsible for

#### 2.4.1.3 Unit Breakdown:



#### 2.4.1.4 Work Breakdown

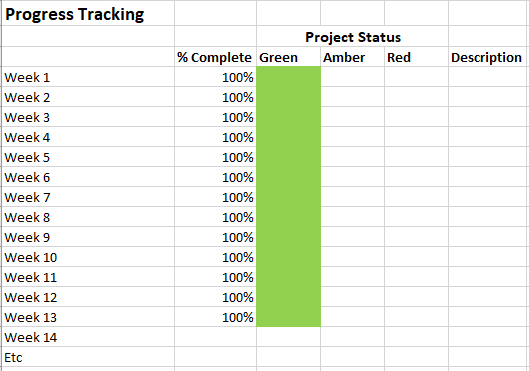
* Design circuit for security inputs (Including Door switches and PIR's)
* Design circuit for output siren (to be run on 12v)
* Design and implement a user interface
* Design and code the SoC to handle all inputs and outputs
* Order parts
* Trade-off decisions
* Assemble the final unit as a whole

#### 2.4.1.5 Timeline

Table : Timeline of Randolph Bock

|  |  |
| --- | --- |
| **Date** | **Description** |
| **17-Jul-17** | **Project Initiation** |
| **10-Aug-17** | **Functional Analysis Test** |
| **14-Aug-17** | **Preliminary Design Complete** |
| **31-Aug-17** | **30% Completion Milestone** |
| **30-Sep-17** | **50% Completion Milestone** |
| **7-Oct-17** | **Design of inputs and outputs Unit 1** |
| **14-Oct-17** | **Code and implementation on SoC** |
| **20-Oct-17** | **Final assembly of Unit 1** |
| **27-Oct-17** | **80% Completion Milestone** |
| **10-Nov-17** | **100% Completion Milestone** |
| **16-Nov-17 to 17-Nov-17** | **Demonstration** |

#### 2.4.1.6 Progress Tracking



### 2.4.2 Experience Report

Myself and two other members has been tasked to design and implement an alarm system for our third years project. The project was divided into 3 units, each member will be responsible for their own unit. The units consist of the physical Alarm Panel, communications, and back-end server. Unit 1 has been allocated to me as I had the most experience in hardware solutions.

The task was daunting at first, but after doing research and determining the requirement, the task then seemed doable within the given timeframe.

Skills required for the task of design Unit 1 the alarm panel is electronic knowledge for the electrical connection required. Programming knowledge to write the embedded system program, as well as knowledge of the Linux operating system distribution, “Raspbian”.

## 2.5 Anton Durandt

## 2.6 FJ Fourie

### 2.6.1 Excel Project Management Documents

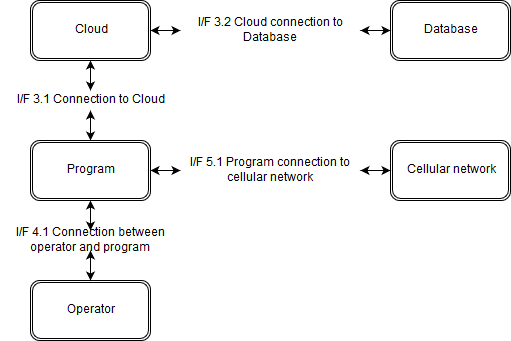
#### 2.6.1.1 Expertise

* Software
* Hardware
* Documentation
* Databases

#### 2.6.1.2 Parts of the project the member is responsible for

* I/F 4 Operator Interface
* I/F 3 Cloud Interface
* I/F 5 Cellular Network

#### 2.6.1.3 Unit Breakdown



#### 2.6.1.4 Work Breakdown

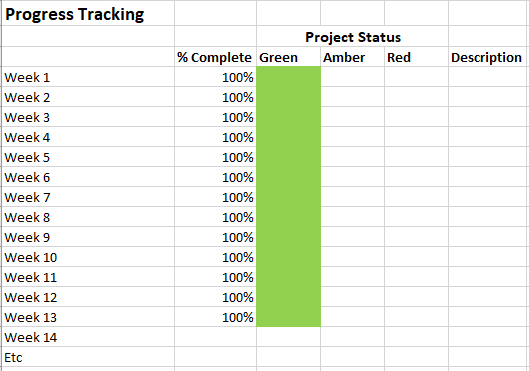
* Design database for information of users and alarm systems
* Design backend operator interface
* Write program for backend of system to manage alarms going off and instruct operators on what to do
* Integrate backend program with database
* Ensure operator interface works correctly with signals received from web
* Assemble the final complete project

#### 2.6.1.5 Timeline

Table : Timeline of FJ Fourie

|  |  |
| --- | --- |
| **Date** | **Description** |
| **17-Jul-17** | **Project Initiation** |
| **10-Aug-17** | **Functional Analysis Test** |
| **14-Aug-17** | **Preliminary Design Complete** |
| **31-Aug-17** | **30% Completion Milestone** |
| **30-Sep-17** | **50% Completion Milestone** |
| **7-Oct-17** | **Design database and program for Unit 3** |
| **14-Oct-17** | **Write program and integrate program with database** |
| **20-Oct-17** | **Final assembly of Unit 3** |
| **27-Oct-17** | **80% Completion Milestone** |
| **10-Nov-17** | **100% Completion Milestone** |
| **16-Nov-17 to 17-Nov-17** | **Demonstration** |

#### 2.6.1.6 Progress Tracking



### 2.6.2 Experience Report

In 2017 which is my 3rd year of studying I have been put into a team of 3 students to complete a project. The project that has been allocated to us is that of an alarm security system we must design and construct the entire system. The project was divided into 3 major units with each member being tasked with completing a unit. The first unit is the alarm panel and user end of the system, the second unit is the communication of the first unit with third unit over the cellular network. The third unit of this project is the backend program that will be connected to a database in the cloud and will be manned by an operator, this is the unit I have been given and tasked with completing. I went about completing this task by firstly identifying all the requirements and individual parts of my unit after which I started researching each component in an attempt to find the best possible solution for each requirement. Through my research and eventual completion of the unit have I broaden my knowledge of the fields of telecommunication, databases, programming and cloud hosting. This has also been a great opportunity for me to have grown as an engineer and thru this process I have grown my skills and abilities in these fields, my knowledge and abilities with the QT programming environment have increased drastically as well as my capabilities in the C++ programming language. I have also learnt a lot about Databases learning to use the program and tools provided by MYSQL to create and host databases. This was also a great opportunity to increase my understanding and abilities with transmission of data over a wireless network using different protocols some of which are existing protocols and others which we designed ourselves. I have also gotten a much broader understanding of the engineering process and all the documentation involved with it as well as all the additional tools used before starting the actual building or programming of the project such as using functional analysis. I have also increased my abilities to work in a team and have learned a lot of important team skills that will be necessary in the work place one day.

# 3 Engineering Methods/Skills/Tools:

## 3.1 Randolph Bock

See additional Document: Design 2017 - Randolph ELO5 - Net1Bier.docx

## 3.2 Anton Durandt

## 3.3 FJ Fourie

See additional Document: Design 2017 - FJ ELO5 - Net1Bier.docx

# 4 Sub-system specification DOCUMENTS:

## 4.1 Randolph Bock

See additional Document: Design 2017 - Randolph Sub-System-specification-Document-Net1Bier.docx

## 4.2 Anton Durandt

## 4.3 FJ Fourie

### 4.3.1 Sub-system functional Analysis

Below in Figure 4 the interface sub system is shown, for the complete interface overview see team design portfolio. Below there is also the Functional analysis of the sub system shown in Figure 5.

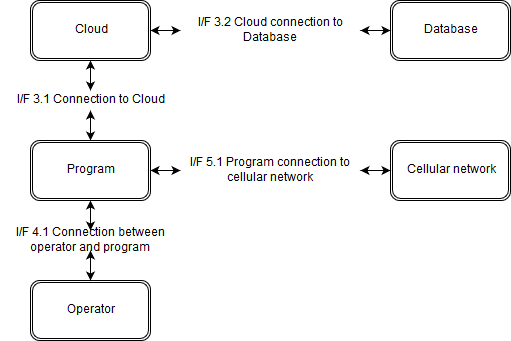


Figure : Interface of sub system

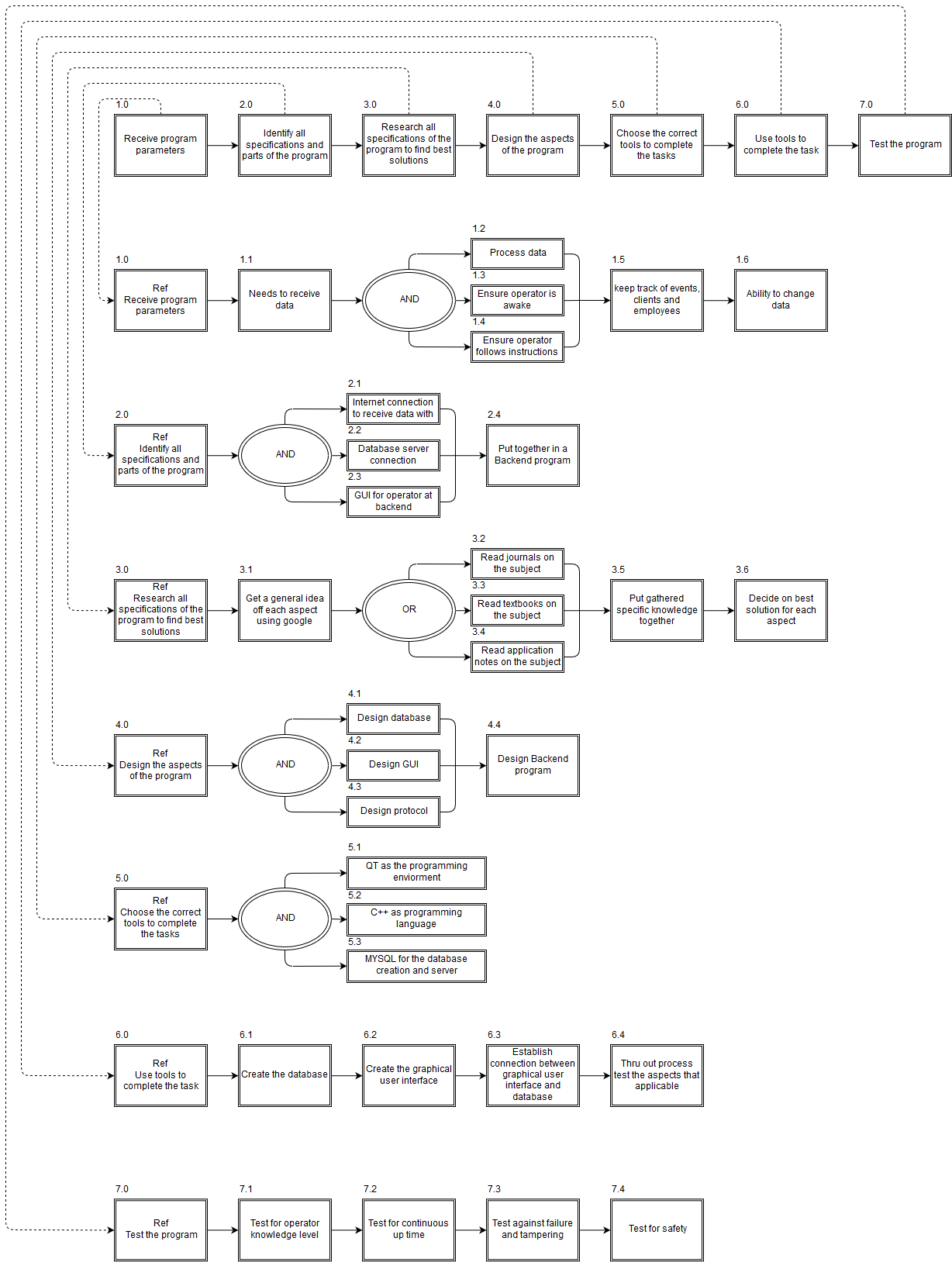


Figure : Functional analysis of sub system

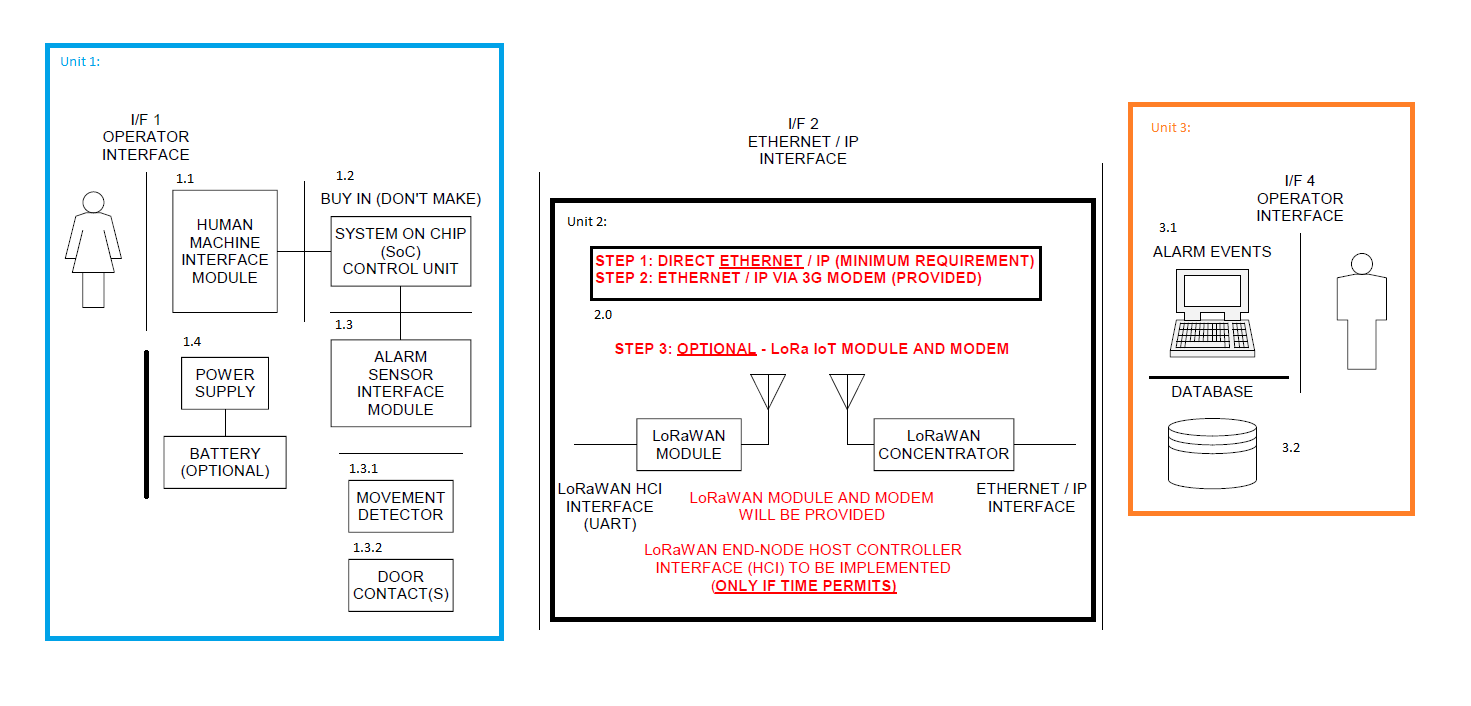
### 4.3.2 Sub-system Interface Definitions

|  |  |  |
| --- | --- | --- |
| **Interface** | **Type** | **Define** |
| I/F 3.1 | Computer | This is the interface between the backend program and the clod server |
| I/F 3.2 | Computer | This is the interface between the cloud server and the database |
| I/F 4.1 | Electronic/Computer | This is the interface between the backend program the operator |
| I/F 5.1 | Electronic/Computer | This is the interface between the back-end program and the cellular network.  The power block will provide 12 V and 6 W power to the alarm panel |
| I/F 2 | System/PC | Network connection between system and PC |

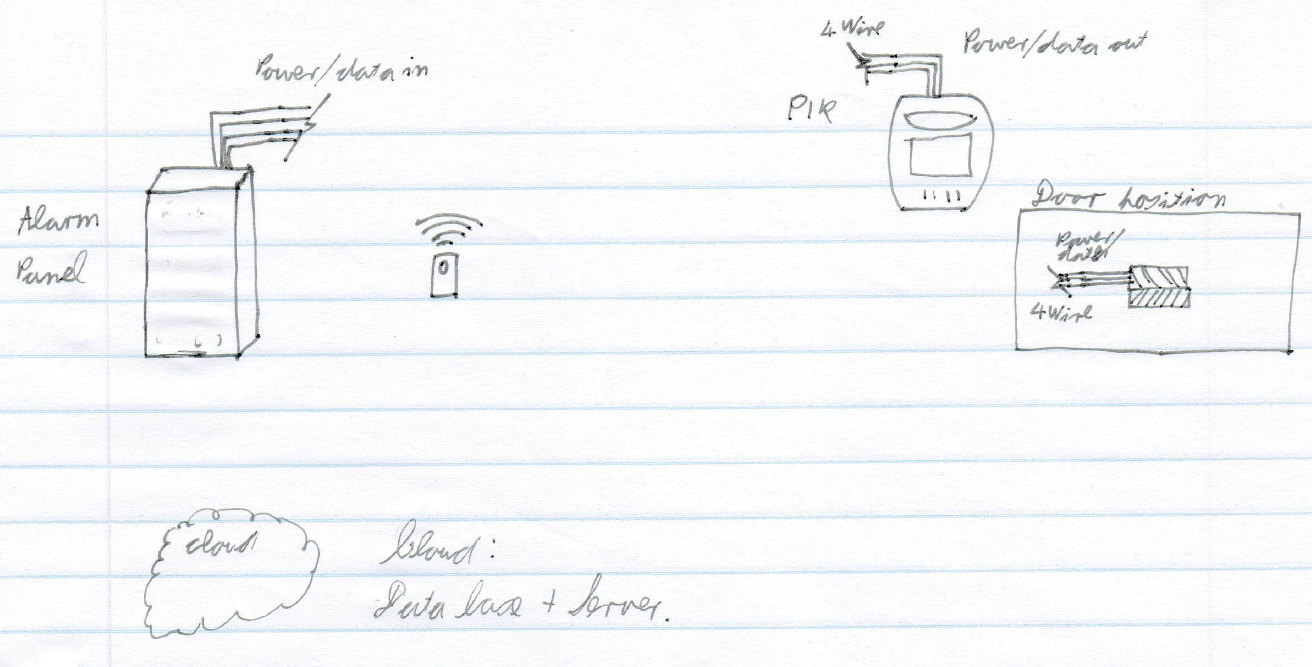
# 5 Design Documentation:

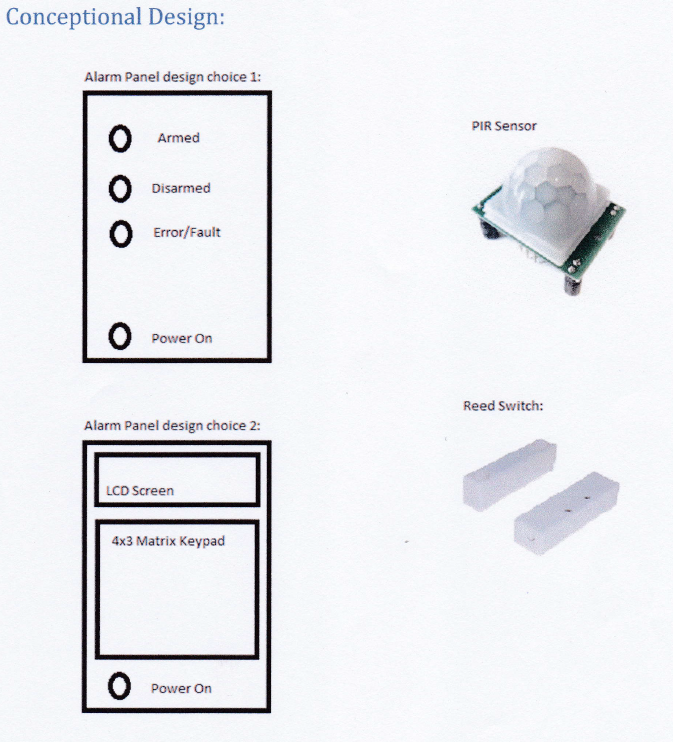
## 5.1 System Design Documaentation

### 5.1.1 Final System Functional Definition:



### 5.1.2 System concept drawings:





### 5.1.3 System interface Definitions:

**Interface control document (I/F 5)**

This interface control document is the interaction between the power block and the alarm.

I/F 5

Power block

Alarm Panel

**Electrical requirements**

The power block will provide 12 V and 6 W power to the alarm panel.

**Mechanical requirements:**

The mechanical interface will be a two-point screw terminal for both sides so that a wire can be connected between the two.

This document was signed in Potchefstroom on the date 2017/09/4 as an agreement between the EERI327/ INEM327 and REII327 students regarding the power supply to the alarm panel. This document is binding until the end of the 3rd year design module 2017.

Representatives

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**C.F. Greyling**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**FJ Fourie**

Witnesses

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Witness 1**



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

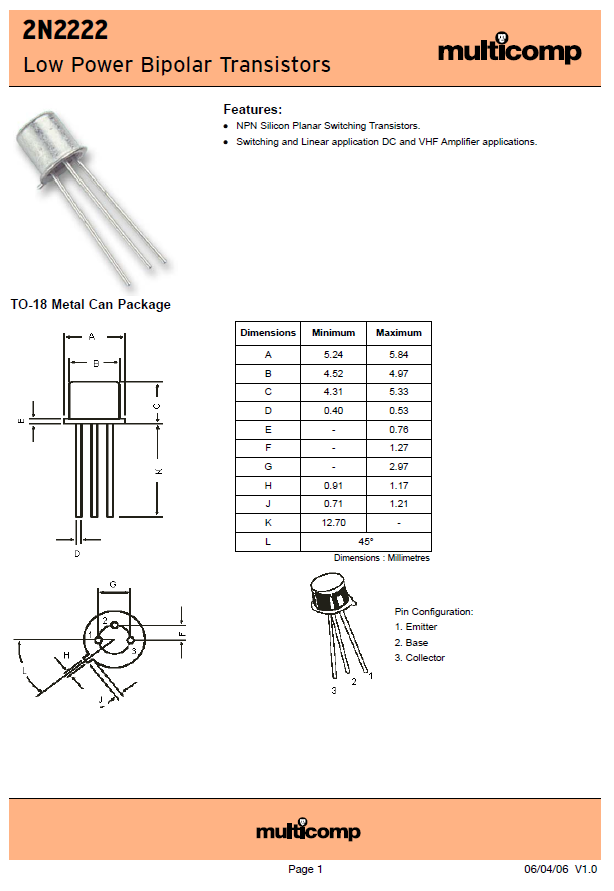
**Witness 2**

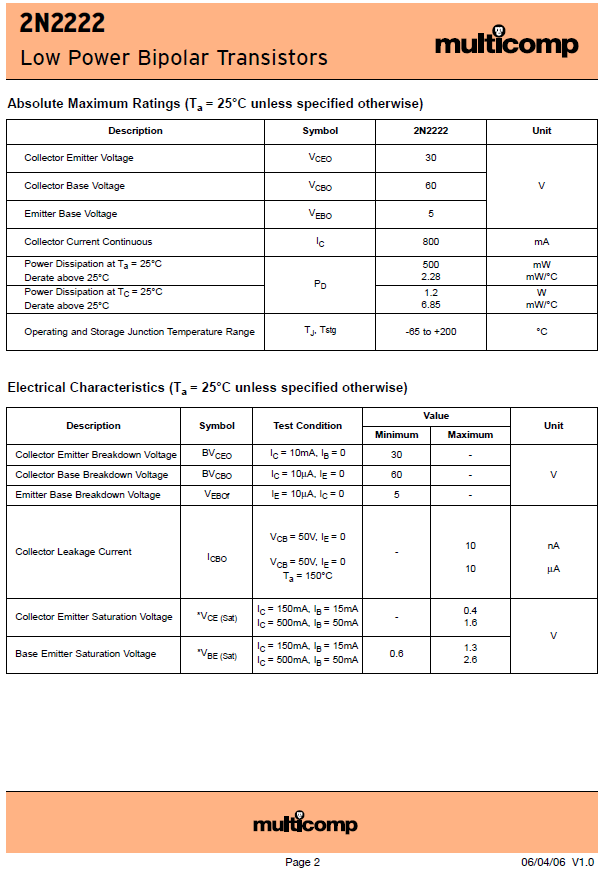
### 5.1.4 System integration testing:

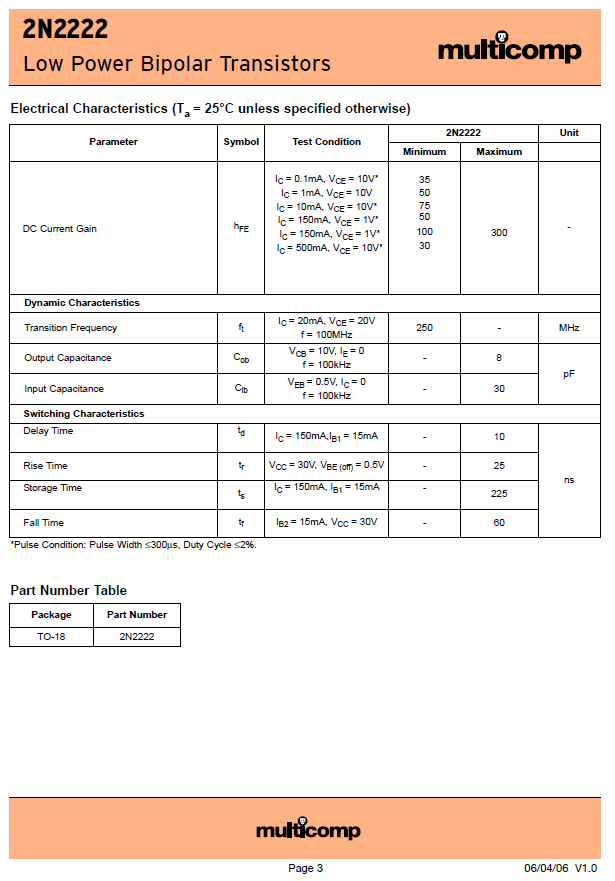
|  |  |  |
| --- | --- | --- |
| Action | Expected Results | ✓/🗶 |
| Trigger Alarm | Message sent through network to PC |  |
| Read from Database | Read specific data from database to PC |  |

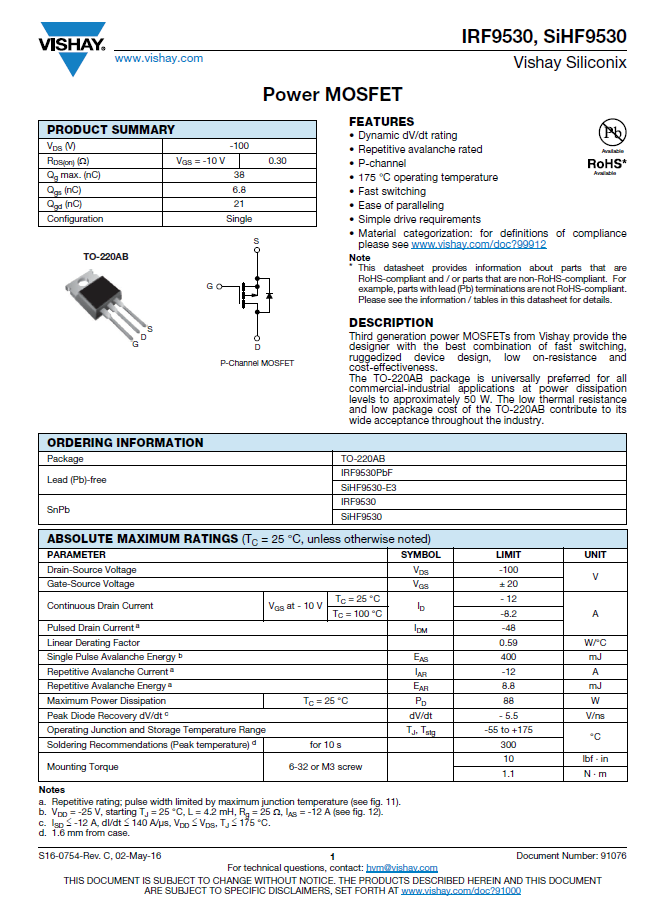
## 5.2 Sub-system Design Documentation Randolph Bock:

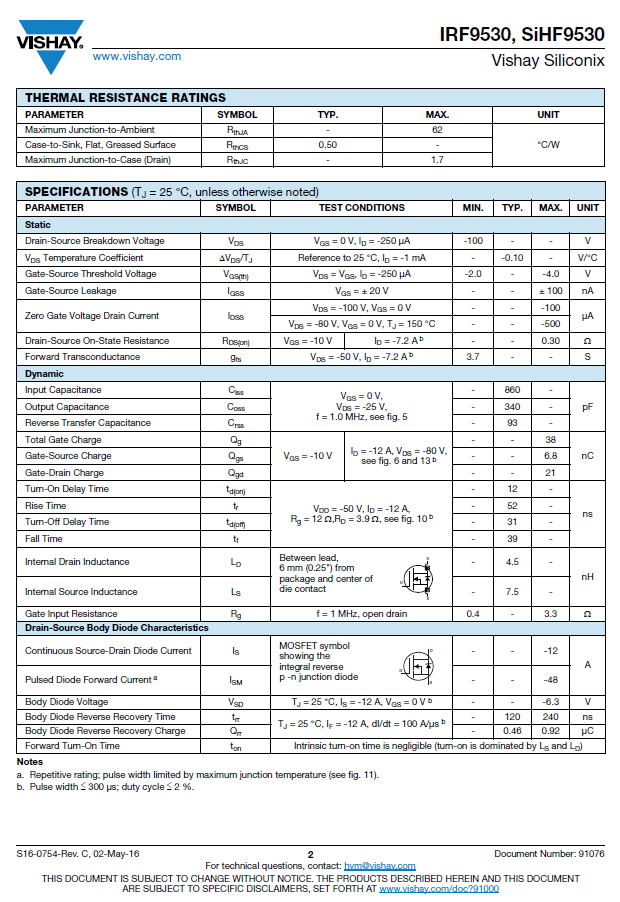
### 5.2.1 Technology Survey/Datasheets

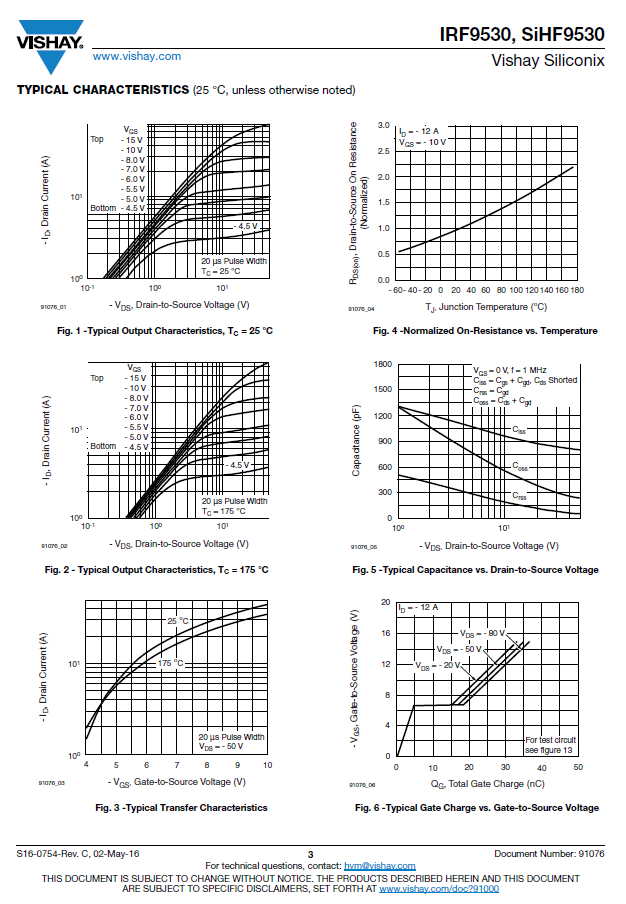


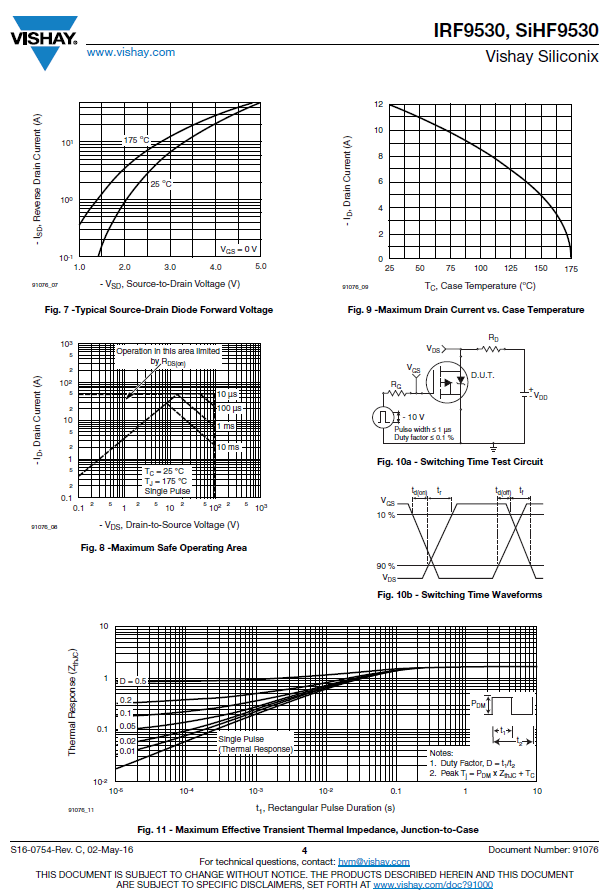


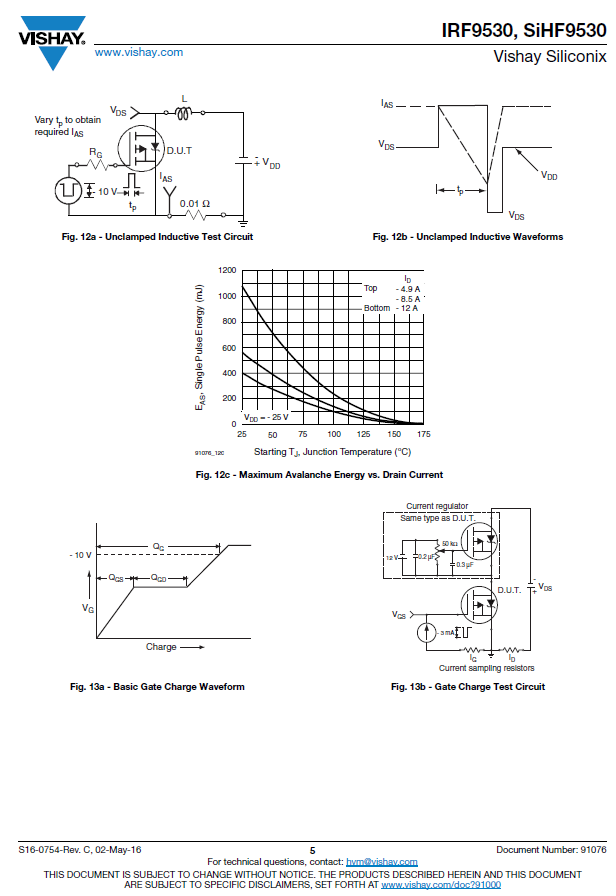


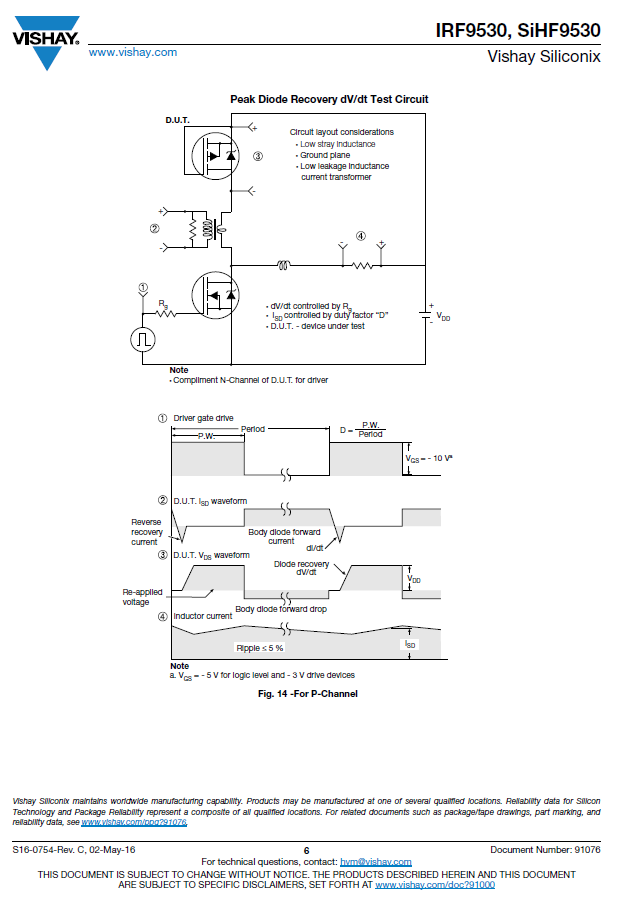


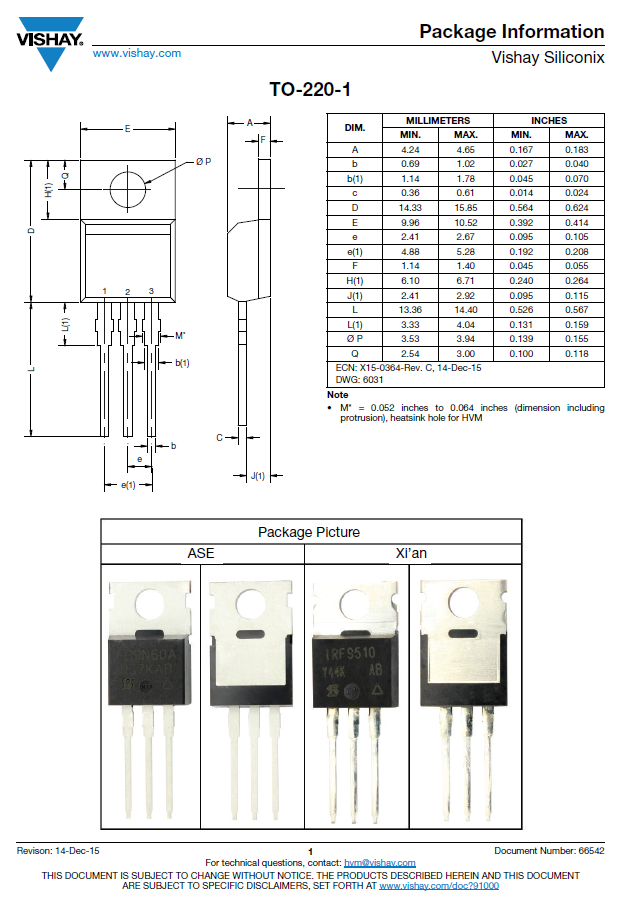


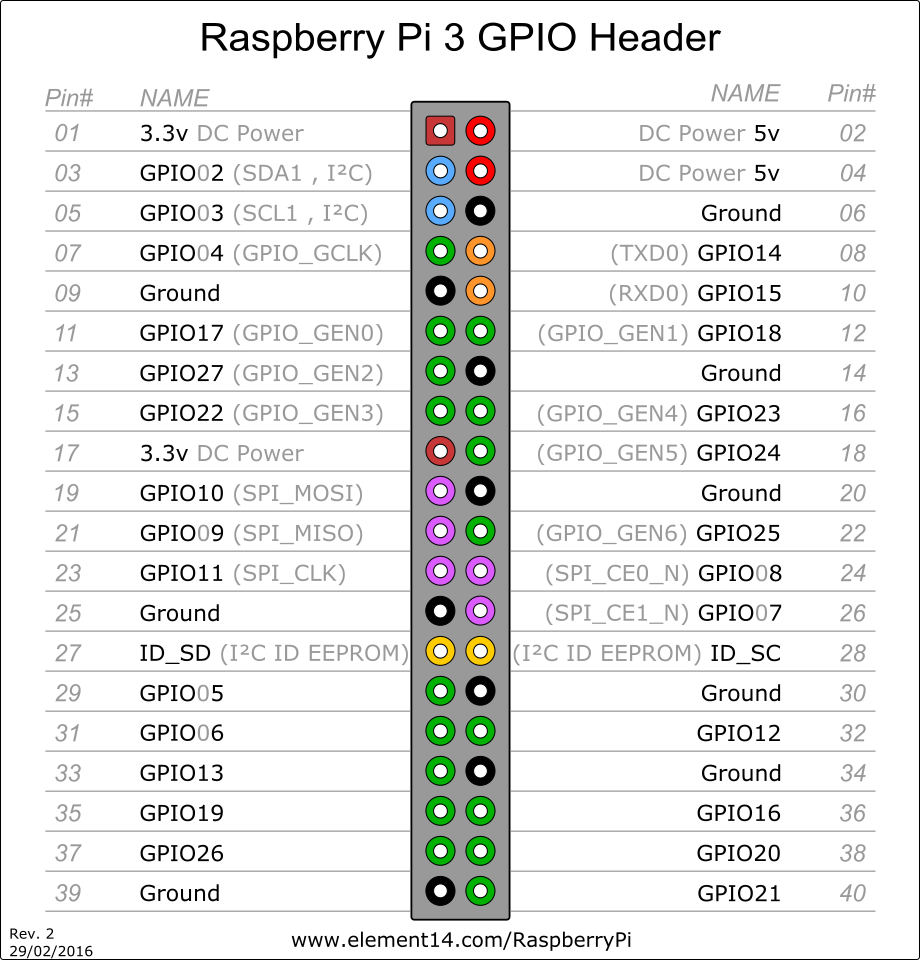












### 5.2.2 Application Notes

GPIO pins can be configured as either general-purpose input, general-purpose output or as one of up to 6 special alternate settings, the functions of which are pin-dependant.

There are 3 GPIO banks on BCM2835.

Each of the 3 banks has its own VDD input pin. On Raspberry Pi, all GPIO banks are supplied from 3.3V. **Connection of a GPIO to a voltage higher than 3.3V will likely destroy the GPIO block within the SoC.**

A selection of pins from Bank 0 is available on the P1 header on Raspberry Pi.

GPIO Pads

The GPIO connections on the BCM2835 package are sometimes referred to in the peripherals datasheet as "pads" - a semiconductor design term meaning "chip connection to outside world".

The pads are configurable CMOS push-pull output drivers/input buffers. Register-based control settings are available for

* Internal pull-up / pull-down enable/disable
* Output [drive strength](http://www.scribd.com/doc/101830961/GPIO-Pads-Control2)
* Input Schmitt-trigger filtering

Power-On States

All GPIOs revert to general-purpose inputs on power-on reset. The default pull states are also applied, which are detailed in the alternate function table in the ARM peripherals datasheet. Most GPIOs have a default pull applied.

Interrupts

Each GPIO pin, when configured as a general-purpose input, can be configured as an interrupt source to the ARM. Several interrupt generation sources are configurable:

* Level-sensitive (high/low)
* Rising/falling edge
* Asynchronous rising/falling edge

Level interrupts maintain the interrupt status until the level has been cleared by system software (e.g. by servicing the attached peripheral generating the interrupt).

The normal rising/falling edge detection has a small amount of synchronisation built into the detection. At the system clock frequency, the pin is sampled with the criteria for generation of an interrupt being a stable transition within a 3-cycle window, i.e. a record of "1 0 0" or "0 1 1". Asynchronous detection bypasses this synchronisation to enable the detection of very narrow events.

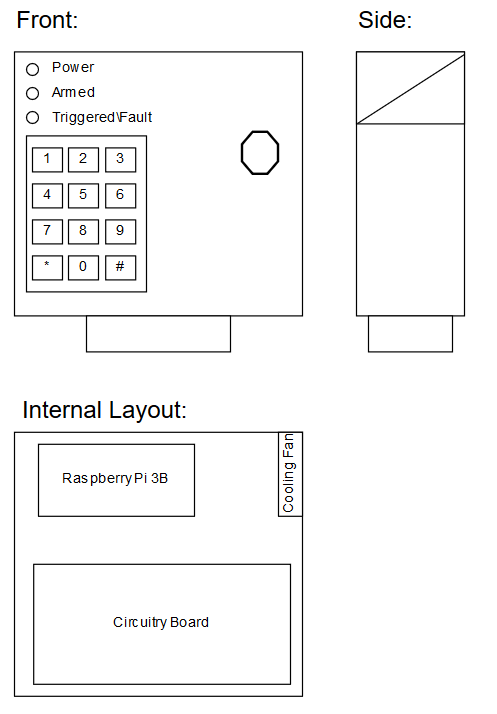
Alternative Functions

Almost all the GPIO pins have alternative functions. Peripheral blocks internal to BCM2835 can be selected to appear on one or more of a set of GPIO pins, for example the I2C busses can be configured to at least 3 separate locations. Pad control, such as drive strength or Schmitt filtering, still applies when the pin is configured as an alternate function.

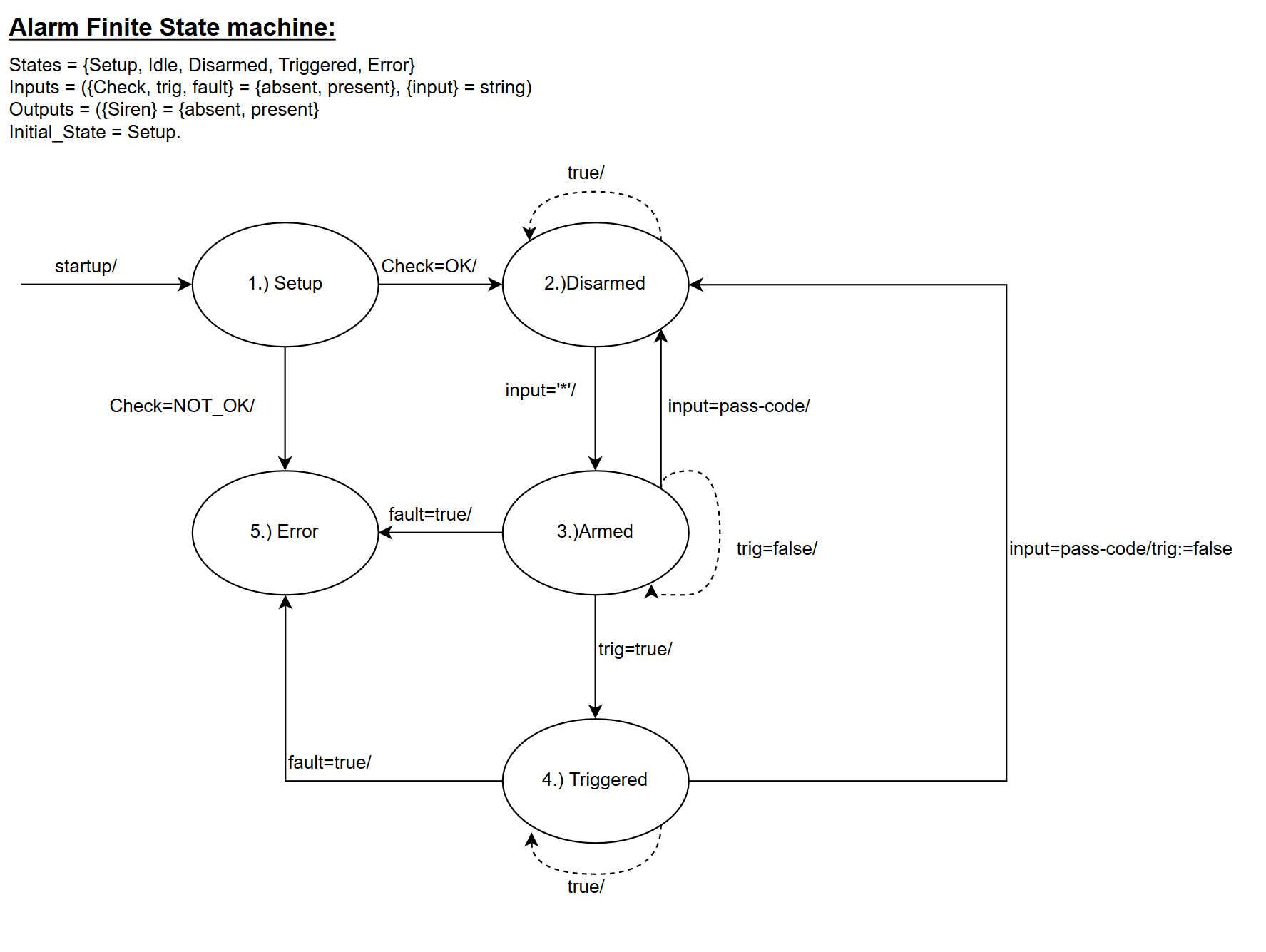
### 5.2.3 Trade off Studies

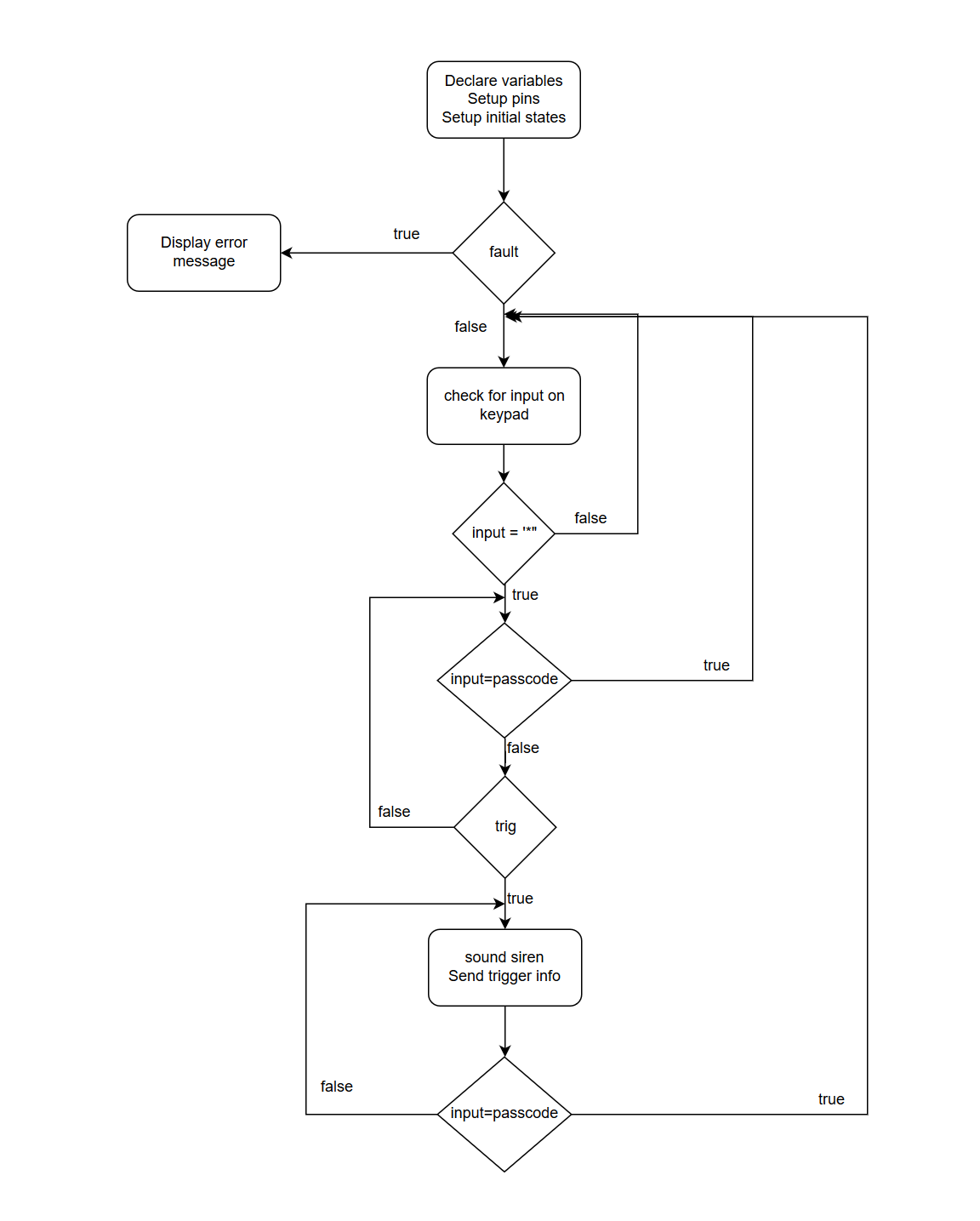
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Visual Human Machine interface (I/F 1):** | | | |  |  |
|  | **LCD Screen** | **LED's** | **7-Segment** |  | **Weight** |
| **Cost** | 3 | 10 | 7 |  | 0,5 |
| **Reliability** | 6 | 9 | 7 |  | 0,3 |
| **Ease of use** | 6 | 7 | 6 |  | 0,2 |
|  | 4,5 | 9,1 | 6,8 |  |  |
|  |  |  |  |  |  |
| **Input Human Machine interface (I/F 1):** | | | |  |  |
|  | **Remote** | **Turn key** | **Keypad** |  | **Weight** |
| **Cost** | 1 | 5 | 9 |  | 0,5 |
| **Reliability** | 2 | 2 | 8 |  | 0,3 |
| **Ease of use** | 7 | 9 | 6 |  | 0,2 |
|  | 2,5 | 4,9 | 8,1 |  |  |
|  |  |  |  |  |  |
| **5V Voltage regulator** | |  |  |  |  |
|  | **LM7805** | **LM371** | **usb1002** |  | **Weight** |
| **Efficiency** | 3 | 4 | 8 |  | 0,5 |
| **Cost** | 8 | 8 | 7 |  | 0,2 |
| **Reliability** | 7 | 8 | 9 |  | 0,3 |
|  | 5,2 | 6 | 8,1 |  |  |

### 5.2.4 Design Drawings:



### 5.2.5 Behavioural Modelling:

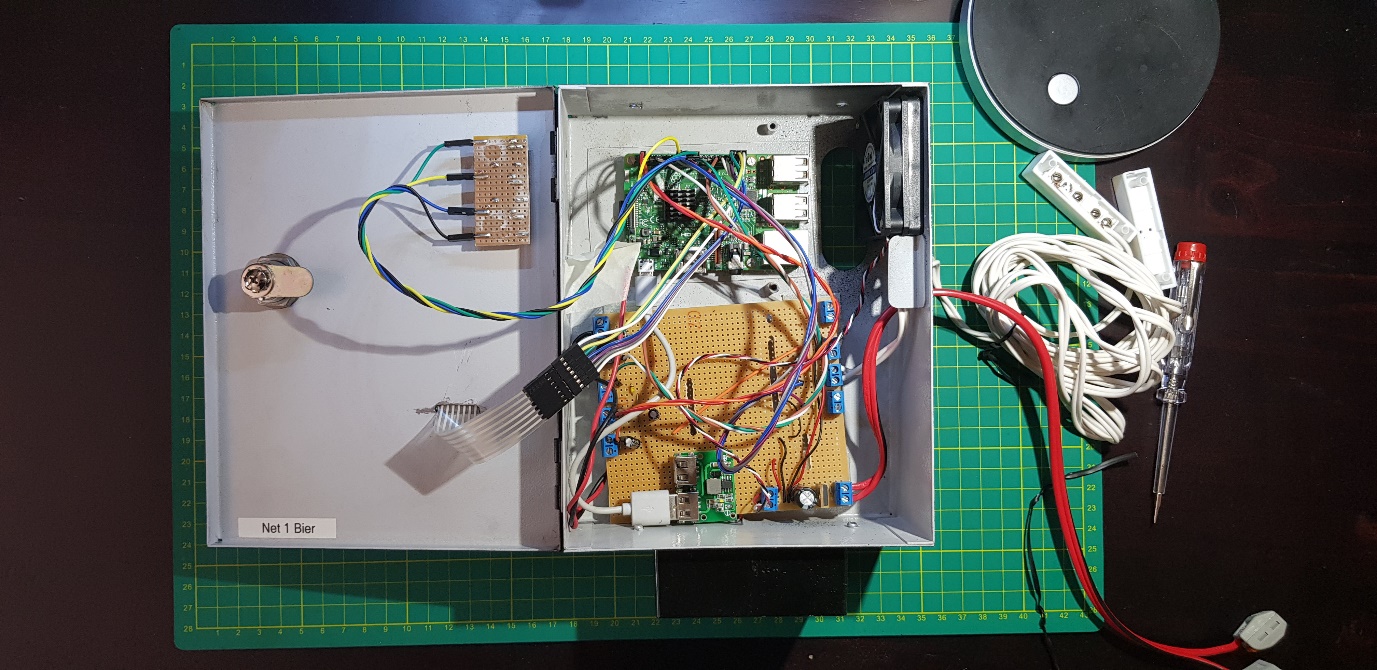




### 5.2.6 Design implementation:

Final construction of Alarm Panel:





### 5.2.7 Sub-system Test and Evaluation:

|  |  |  |
| --- | --- | --- |
| Action | Expected Results | ✓/🗶 |
| Program started | Terminal shows program has started and heartbeat signal is shown on the power status LED | ✓ |
| Buttons are Pressed on the Keypad | Terminal shows the button pressed | ✓ |
| The “\*” button is pressed in idle mode to enter armed Mode | The terminal displays that the system is arming, when the system is armed the terminal will display armed, and a heartbeat signal will show on the Armed status LED | ✓ |
| Moving the reed switch while in armed mode | The siren starts ringing. The terminal displays a trigger on the Reed sensor. The Triggered status LED starts flashing | ✓ |
| Moving the PIR sensor while in armed mode | The siren starts ringing. The terminal displays a trigger on the PIR sensor. The Triggered status LED starts flashing | ✓ |
| Simulating an input outdoor sensor signal while in armed mode | The siren starts ringing. The terminal displays a trigger on the Outdoor sensor. The Triggered status LED starts flashing | ✓ |

|  |  |  |
| --- | --- | --- |
| Action | Expected Results | ✓/🗶 |
| Power up | The system run through the setup phase and the heartbeat signal is shown on the power status LED. The system is then in the idle state. |  |
| The “\*” button is pressed in idle mode to enter armed Mode | A buzzer will start sounding and after 5 second the system will enter the armed state. A heartbeat signal will show on the Armed status LED |  |
| Moving the reed switch while in armed mode | The siren starts ringing. The Triggered status LED starts flashing. A UDP message is sent showing a Door switch trigger has occurred |  |
| Triggering the PIR sensor while in armed mode | The siren starts ringing. The Triggered status LED starts flashing. A UDP message is sent showing a PIR sensor trigger has occurred |  |
| Triggering the PIR sensor while in armed mode | The siren starts ringing. The Triggered status LED starts flashing. A UDP message is sent showing an outdoor sensor trigger has occurred |  |
| Enter the passcode + “#” while in armed state | The system turns of the armed status LED and returns to the idle state |  |
| Enter the passcode + “#” while in triggered state | The system turns of the triggered status LED and returns to the idle state |  |