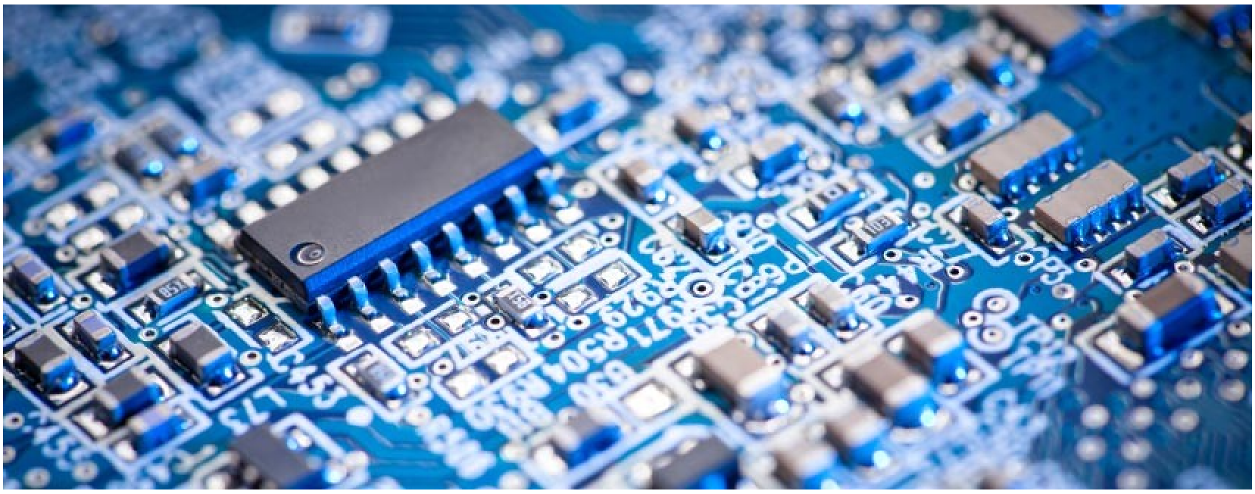
Electronic Design 314
15/05/2015

Ignatius de Villiers – 17502292
Lian Malan – 17515734

Final Report



INDEX

1. DECLARATION......2
2. SUMMARY......3
3. INTRODUCTION......4
4. SYSTEM DESCRIPTION......5
5. DETAIL DESIGN......6
   Buzzer......6
   Power Supply Sensor......7
   IT de Villiers......8
      HALT Modes Total Current......8
      Current Measurement (in FULL mode)......9
      Sensor Measurements......9
      Pressure Coefficients Values......9
   L Malan......10
      HALT Modes Total Current......10
      Current Measurement (in FULL mode)......10
      Sensor Measurements......11
      Pressure Coefficients Values......11
6. PROGRAMS......12
      User Defined Function Descriptions......12
      All Project Files......13
      Libraries Included......15
7. CONCLUSION......20
8. LIST OF ABBREVIATIONS......21
9. REFERENCES......21

APPENDICES

A USER MANUAL

B TECHNICAL SPECIFICATION

C CIRCUIT DIAGRAM

1. Declaration

We the undersigned, hereby declare that this report and the work contained therein is our own original work, except where indicated.

Signatures:

Ignatius de Villiers	Date
Lian Malan	Date

2. Summary

This report provides detailed information on an environment monitoring system that tracks a number of environmental conditions. The monitoring system provides accurate real-time measurements of temperature, humidity, air pressure, wind speed and date and time. It also allows human and computer interaction via a simple 5 button keypad, LCD, buzzer-alarm and UART communication. These measurements and interactions, are made possible by intricate circuitry and C-code. All calculations and circuit diagrams can be found in the appendices.

Results of data analysed, show that this system can provide the user with accurate real-time temperature, humidity, air pressure, wind speed and date and time measurements that can be stored and analysed by the user at a later stage.

The report finds that this system is a user friendly and interactive system that can accurately monitor environmental conditions in real-time, which can be best implemented in environments where temperature, humidity, air pressure and wind speed play a big role in human or general safety.

Recommendations discussed include:

- Protective casing for the system.
- Enabling a Bluetooth connection with the system.
- Adding a Power on-and-off button.
- Adding a Reset button.
- Using a Louder Buzzer.

The report also investigates the fact that the system may have some limitations

Limitations include:

- Will not function correctly under unnatural conditions.

- Cannot detect smoke.
- Cannot use in wet conditions

3. Introduction

The system that we built, is designed to measure certain weather conditions. These include the temperature, relative humidity, air pressure and the wind speed. The wind speed measurement is done +sing a mechanical component and interrupt signals. We use IIC communication to retrieve the other measurements.

A main supply voltage of 9V is used to power the system, with an optional backup battery supply available. The system will be fully functional in either case. The system will always keep track of which power supply is currently being used.

A button system is available so that the user can interact with the system and access certain functions, ex. Viewing the alarm warnings and displaying the measured values.

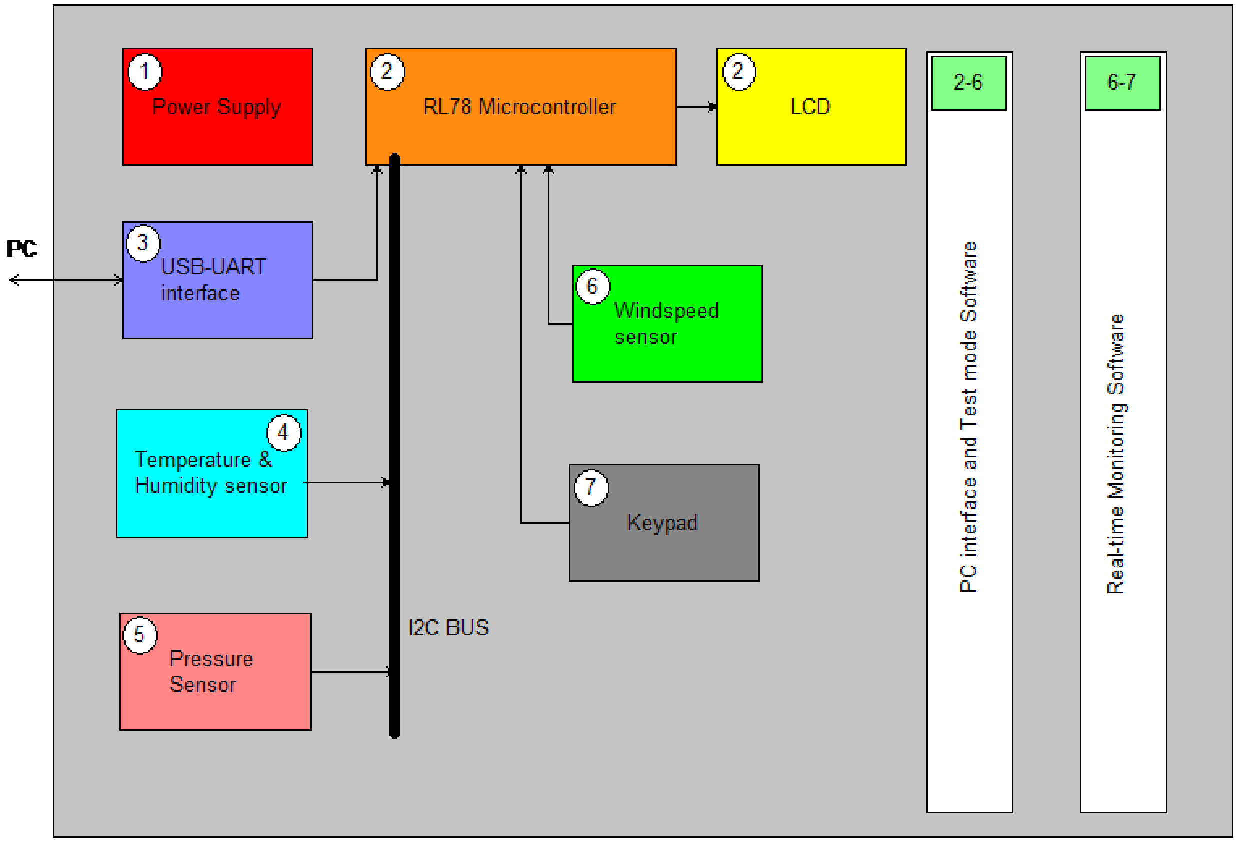
The system includes a real time clock so that the user can see, or edit the current date and time. It is also used to keep track of when the alarm warnings took place.

A buzzer will be used to indicate that certain functions are being accessed, and to alert the user that an alarm condition has been met, ex. That the temperature value is too high.

A 16-bit LCD screen is implemented so that the system can communicate with the user. There is an added function which allows the user to display a text message on the screen.

Each of these implemented functions were designed and built separately and connected to the RL78 Microcontroller, that allows the different components to communicate with each other and work together to achive our final product's goal.

4. System Description



See appendix C

1) Supplies the entire system with power.

2) The system communicates with its environment through the RL78 Microcontroller, LCD and relevant C-code.

3) Allows serial communication over a computer. Translates data between parallel and serial forms.

4) Allows Temperature and Humidity sampling.

5) Allows Pressure sampling.

6) Allows wind speed sampling.

7) Allows a more interactive user friendly Interface.

5. Detail Design

Buzzer

We used the MT12G_2P Buzzer in our system. The buzzer circuit is shown in Appendix C, Figure 2. The picture extract (below) from the data sheet of the buzzer, specifies that it has a voltage limit of 2V and a current limit of 10mA.

When the transistor is on, the 10MΩ resistor serves as a ground connection for the current of the transistor. This resistor is chosen sufficiently large.

A diode is placed in parallel with the buzzer, to protect the buzzer from large spikes in voltage.

The 270Ω resistor is used to limit the current through the buzzer. The calculations for this resistor are as follow:

The transistor (2N7000) data sheet states that for V_{GS} = 4.5V and I_D = 75mA, V_{DS}(max) = 0.4V.

$V_{BUZZ/NEG}$ = VCC - V_{BUZZER} = 5 - 2 = 3V (At the negative pin of the buzzer)

I_D has to be smaller than 10mA.

Therefor:

$$\frac{V_{BUZZ/\neg\text{¿}} - V_{DS(max)}}{R\ \text{¿}} < V_{BUZZ(max)}$$

$$\frac{3-0.4}{R} < 0.1$$

R > 260Ω

We chose: R = 270Ω

Part No.	Stock Code	Operating Voltage (VDC)	Rated Current (mA)	Min.Sound Output (dBa/10cm)	Size (Dia x H)	Price (HK$ / Pcs)
MT12G-1P	267-066	1 ~ 2	30	85	12 x 8.5 mm	3.00
MT12G-2P	267-085	1 ~ 2	10	80	12 x 8.5 mm	3.00
MT12-105B	267-003	1 ~ 2	10	70	12 x 5.4 mm	4.00
MTW09B	267-088	2 ~ 6	80	82	9 x 5.5 mm	3.40

EXTRACT

Power Supply Sensor

This circuit determines whether the system is currently receiving power from the main supply or the battery. The circuit can be seen in Appendix C, Figure 10.

The resistance values has to be calculated. We look at the data sheet of the transistor used: 2N3904

For the transistor in the on state:

30 <= h_{FE} <=300

h_{FE} is chosen to be a minimum.

$V_{CE(sat)}$ = 0.2V and $V_{BE(sat)}$ = 0.65V,

for I_C = 10mA and I_B = 1mA (Minimum values).

When the input voltage is higher than 6V (battery voltage), a logical high is sent to the processor.

$$I_C = \frac{V cc - V ce(sat)}{Rc}$$

$$I_C = \frac{5 - 0.2}{Rc}$$

V_{input} - $V_{BE(sat)}$ = 6 - 0.65 = 5.35V

$$R_B \leq \frac{V_{RB}}{\frac{I_C}{h_{FE}}} = \frac{5.35}{\frac{4.8}{R_C}} \times 30 = 33.4375R_C$$

We chose R_C = 4.8kΩ,

Therefor R_B ≤ 160.5kΩ.

We chose R_B = 69.5kΩ (A safe choice)

NOTE: The rest of the components (resistors, capacitors, etc.) was either specified in the data sheet of the certain component, or it was chosen to be sufficiently high/low.

Measurements and Results

The following measurements were all done in the Second floor lab in the Engineering Faculty of Stellenbosch University on 14/05/2015

IT de Villiers

Main

5V Regulator: 4.96V

3.3V Regulator: 3.28V

Battery

5V Regulator: 4.96V

3.3V Regulator: 3.28V

HALT Modes Total Current

No HALT (FULL)

Main: 20.9mA

Battery: 19.7mA

Always HALT (HALT)

Main: 16.67mA

Battery: 15.5mA

Battery Only HALT (Default)

Main: 20.9mA

Battery: 15.5mA

Current Measurement (in FULL mode)

Buzzer On

Main: 29.2mA

Battery: 28mA

Button Pressed

Main: 21.45mA

Battery: 20.3mA

Sensor Measurements

Temperature Sample: 22 °C

Humidity Sample: 55.5%

Pressure Sample: 100.5 kPa

Pressure Coefficients Values

a0 = 17'902 (mixed signed 16.3 format)

b1 = -21'665 (mixed signed 16.13 format)

b2 = -18'169 (mixed signed 16.14 format)

c12 =15'965 (24-bit number with 13-bit fraction and 9 padded zeroes)

L Malan

Main

5V Regulator: 4.98V

3.3V Regulator: 3.31V

Battery

5V Regulator: 4.98V

3.3V Regulator: 3.31

HALT Modes Total Current

No HALT (FULL)

Main: 21.6mA

Battery: 20.5mA

Always HALT (HALT)

Main: 17.55mA

Battery: 16.2mA

Battery Only HALT (Default)

Main: 21.7mA

Battery: 16.2mA

Current Measurement (in FULL mode)

Buzzer On

Main: 30mA

Battery: 28.7mA

Button Pressed

Main: 22.25mA

Battery: 20.9mA

Sensor Measurements

Temperature Sample: 23degC

Humidity Sample: 53%

Pressure Sample: 100kPa

Pressure Coefficients Values

a0 = 18'962 (mixed signed 16.3 format)

b1 = -22'964 (mixed signed 16.13 format)

b2 = -19'139 (mixed signed 16.14 format)

c12 = 16'376 (24-bit number with 13-bit fraction and 9 padded zeroes)

6. Programs

User Defined Function Descriptions

File	Function	Description
my_buzzer.c	BUZZ_LOOP()	Two tone alarm buzz sequence. (i) alarm on for 5 seconds or (ii) on till middle button is pressed
	Buzzer()	Buzzer plays appropriate buzz sequence according to "buzState".
my_functions.c	alarm_new()	When there is a new alarm, this function displays the alarm regardless of mode (Normal / Test).
	alarms_menu()	Displays appropriate error message in Normal mode.
	buttons()	Allows buttons to perform specific menu tasks during Normal and test mode.
	cycle()	Allows the periodic display of temperature, humidity, pressure, wind speed and date & time during normal mode.
	flag_check()	Evaluates whether interrupt flags have been set – then executes the suitable task in accordance to the flag condition.
	Read_status()	Determines which alarm conditions are currently being triggered. Ex. If current temperature vale > max temperature alarm value
	values_menu	Manage whether to display menu option (temperature, humidity,

		pressure, wind speed or date & time) or actual value during normal mode.
my_lcd.c	clearBuff(int)	Clears the RX buffer.
	Delay(int n)	Supplies an n second(s) delay.
	delayNoInt(uint16_t n)	Supplies an n micro second(s) delay.
	initializedlcd()	Initialize the LCD to a point where it can start displaying data.
	LCD_shift_display()	Allows automatic scrolling through a message on the LCD.
	LCD_write_Byte(char)	Writes a byte (8-bits) to the LCD, and displays it.
	LCD_write_Nibble(char)	Writes a Nibble (4-bits) to the LCD, and displays it.
	powerCheck()	Determines whether main or secondary power source is connected and displays an appropriate message.
	Start(char[], int)	Display char array on the LCD.
	writeNumber(char*,char)	Write a number on the LCD.
	writeNumber2(char*,char)	Write date and time – numbers on LCD.
r_cg_rtc_user.c	BCD_to_Dec(uint8_t)	Converts binary coded decimal to decimal.
	Dec_to_BCD(char)	Converts decimal to binary coded decimal.
r_cg_serial_user.c	calCoef(char*)	Calculate pressure coefficients used for pressure value calculations.
	calcPComp(uint16_t, unint16_t)	Calculate pressure value.
	calWind()	Calculate wind speed to be displayed on LCD.

	convertHumid(char*)	Calculate humidity value to be display on LCD.
	convertTemp(char*)	Calculate temperature value to be display on LCD.

All Project Files

Generated Files	Pre-Built Files	User Defined Files
r_reset_program.asm	iodefine_ext.h	my_buzzer.c
r_cg_cgc_user.c	iodefine.h	my_buzzer.h
r_cg_cgc.c	ED314_Project1.c	my_functions.c
r_cg_cgc.h	hardware_setup.c	my_functions.h
r_cg_intc_user.c	interrupt_handlers.c	my_lcd.c
r_cg_intc.c	interrupt_handlers.h	r_main.c
r_cg_intc.h	reset_program.asm	
r_cg_interrupt_handlers.h	typedefine.h	
r_cg_macrodriver.h	vector_table.c	
r_cg_pclbuz_user.c		
r_cg_pclbuz.c		
r_cg_pclbuz.h		
r_cg_port_user.c		
r_cg_port.c		
r_cg_port.h		
r_cg_rtc_user.c		
r_cg_rtc.c		

r_cg_rtc.h		
r_cg_serial_user.c		
r_cg_serial.c		
r_cg_serial.h		
r_cg_timer_user.c		
r_cg_timer.c		
r_cg_timer.h		
r_cg_userdefine.h		
r_cg_vector_table.c		
r_hardware_setup.c		

Libraries Included

```
#include <string.h>
#include <stdlib.h>
```

Temperature calculations:

$$T = -46.85 + 175.72 \times \frac{S}{2^{16}}$$

C-code for Temperature calculations:

```
char ans = ((175.72)*(rawT[0]))/(1U<<8)-46.85;
return (2*ans);
```

Humidity calculations:

$$RH = -6 + 125 \times \frac{S}{2^{16}}$$

C-code for Humidity calculations:

```
uint32_t ans = ((250U*(rawH[0]))>>8)-12;

return ((uint8_t)ans);
```

Wind speed calculations:

v(k) = 0.8 [v(k − 1) + 0.25 (p(k) − p(k − 1))]

C-code for Wind speed calculations:

```
WindOldTurns = wind_count1 + wind_count2;

wind_count1 = 0;

wind_count2 = 0;
```

Air pressure calculations:

Pcomp = a0 + (b1 + c12 x Tadc) x Padc + b(c)
Pressure(kPa) = Pcomp [115-50/ 1023] + 50

C-code for air pressure calculations:

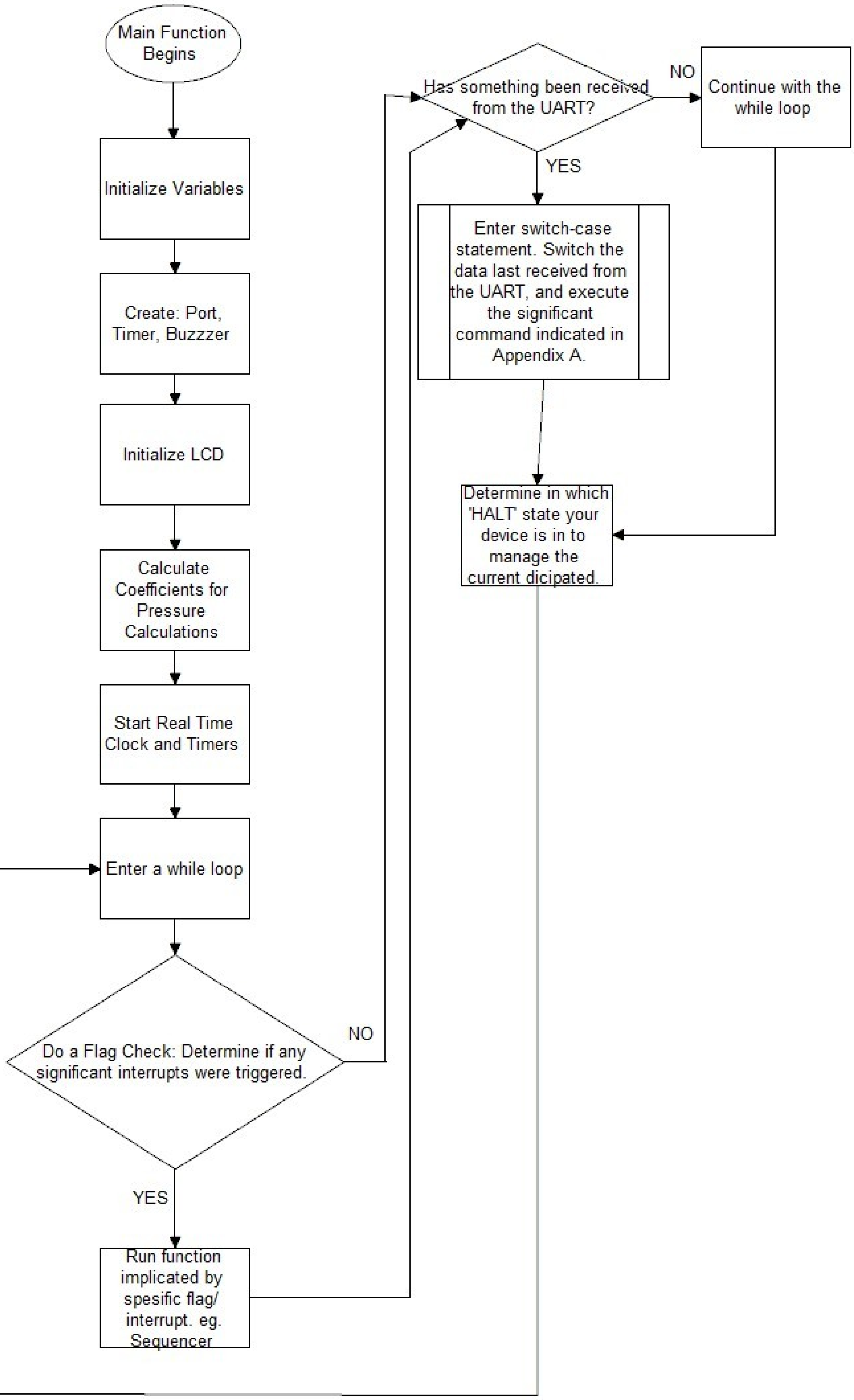
```
int32_t c12x2, a1x1, a1, y, a2x2, pCompScaled, y1;
        uint16_t pInt, tInt;
        /* Temperature and pressure in fractional format (not correctly stated
         * in the data sheet), so we need to make them integers first. The
         * resolution is 10 bits, so we can simply right-shift them by 6 */
        pInt = mplPadc >> 6;
        tInt = mplTadc >> 6;
        /* c12 is in mixed 24.13 format, by shifting the product c12*T right by
         * 11 bits, the answer will be in the same format as b1*/
        c12x2 = ((int32_t) mpl115Coef_c12 * tInt) >> 11;
        /* a1 is now (b1 + c12*T) */
        a1 = (int32_t) mpl115Coef_b1 + c12x2;
        /* a1x1 is now (b1 + c12*T)*P */
```

```
a1x1 = a1 * pInt;
/* a1x1 is in mixed 32.13 format, so a0 must be shifted left by 10 bits
 * y1 is now a0 + (b1 + c12*T)*P */
y1 = ((int32_t) mpl115Coef_a0 << 10) + a1x1;
/* b2 is in mixed 16.14 format, shift by 1 left to get to 32.13 format
 * a2x2 is now b2*T*/
a2x2 = ((int32_t) mpl115Coef_b2 * tInt) >> 1;
/* We need pComp with enough resolution, so shift right by 9 bits
 * (instead of 13 bits so the value returned will be 16 times bigger
 * than the real value)
 * pComp is now y1 + a2x2 = (a0 + (b1 + c12*T)*P) + (b2*T)
 * pCompScaled = 16*pComp4*/
pCompScaled = (y1 + a2x2) >> 9;
/* Just use the lower 16 bits, by limiting the range */
if (pCompScaled > 20000)
        pCompScaled = 20000;
else if (pCompScaled < 1)
        pCompScaled = 1;
return((uint16_t) pCompScaled);
```

Sequencer Flow (Timer Delays)

See below the two main flow diagrams

Main Function Flow


Main Function Begins
Initialize Variables
Create: Port, Timer, Buzzzer
Initialize LCD
Calculate Coefficients for Pressure Calculations
Start Real Time Clock and Timers
Enter a while loop
Do a Flag Check: Determine if any significant interrupts were triggered.
NO
YES
Run function implicated by spesific flag/ interrupt. eg. Sequencer
Has something been received from the UART?
NO
Continue with the while loop
YES
Enter switch-case statement. Switch the data last received from the UART, and execute the significant command indicated in Appendix A.
Determine in which 'HALT' state your device is in to manage the current dicipated.

7. Conclusion

In conclusion it is clear that this environment monitoring system is a user friendly and interactive system that can accurately monitor environmental conditions in real-time, simultaneously allowing the user to interact with the system through a computer or via the keypad. This system can be best implemented in environments where temperature, humidity, air pressure and wind speed play a big role in human or general safety.

This system can however be improved by adding the following:

- Protective casing for the system, to ensure normal functioning during wet conditions.
- Adding a smoke detector could improve functionality of the system as it could set smoke alarms off, which could prevent severe fire damage.
- Adding Bluetooth communication could mean that a user could see real time measurements and alarms produced by the system, remotely read and erase the data log or set alarm values on a phone or PC.
- Users may also want to save battery life if there is no necessity for the system, thus adding an on/ off button would increase functionality.
- Adding a Reset button could also increase the systems functionality.
- In noisy areas a louder buzzer or speaker might be necessary to hear alarms.

The system does have limitations which could be problematic in some situations.

Limitations include:

- The system might malfunction in un-naturally hot or cold conditions as the SHT21 has max and minimum temperature limitations and some circuit elements are sensitive to heat/cold.

- Although the system can detect temperature, it cannot detect smoke, which might mean that fire detection might not be as accurate as a user would like.
- As the system is completely electronic, exposure to wet conditions could cause water damage, and complete system failure.

8. List of abbreviations

Abbreviation	
LCD	Liquid crystal display
UART	Universal asynchronous receiver/transmitter
USB	Universal serial bus
PC	Personal Computer
Ex.	Example

1. References

1. http://www.abbreviations.com
2. http://www.kbodynamics.co.uk/images/image/slides/opt/kbo-5.jpg
3. http://learn.sun.ac.za/pluginfile.php/272793/mod_resource/content/1/r01uh0146ej0300_rl78g13.pdf
4. http://learn.sun.ac.za/pluginfile.php/327050/mod_resource/content/2/TS2940.pdf
5. http://learn.sun.ac.za/pluginfile.php/327051/mod_resource/content/1/1n4001.pdf
6. http://learn.sun.ac.za/pluginfile.php/327053/mod_resource/content/3/lp2950-n.pdf
7. http://learn.sun.ac.za/pluginfile.php/335246/mod_resource/content/1/2N3904_FC.pdf
8. http://learn.sun.ac.za/pluginfile.php/335400/mod_resource/content/1/275FXXX%20Fuze%20Mantech%2014M1195.pdf
9. http://learn.sun.ac.za/pluginfile.php/326966/mod_resource/content/2/DS_FT230X.pdf
10. http://learn.sun.ac.za/pluginfile.php/352179/mod_resource/content/1/Sensirion_Humidity_SHT21_Datasheet_V4.pdf
11. http://learn.sun.ac.za/pluginfile.php/354985/mod_resource/content/1/2N7000.pdf
12. http://learn.sun.ac.za/pluginfile.php/361834/mod_resource/content/2/MPL115A2.pdf
13. http://learn.sun.ac.za/pluginfile.php/360980/mod_resource/content/1/MT12G_2P.pdf
14. http://learn.sun.ac.za/pluginfile.php/360984/mod_resource/content/4/ALPS%20EC12E2420404%20Specification.pdf

Appendix A

User Manual

Test Mode Instructions	
Hex Value	Command
0xF0	Buzzer ON
0xF1	Buzzer OFF
0xF2	Set Clock
0xF3	Send current clock data
0xF4	Display ASCIIZ String on LCD
0xF5	Erase data Buffer
0xF6	Send data Buffer to UART
0xF7	Read Temperature
0xF8	Read Humidity
0xF9	Read Pressure
0xFE	Read Windspeed
0xFF	Read status
0x80	Switch to Normale mode
0x81	Switch toTestmode

Appendix B

Technical Specification

Power

The system requires an external DC Power Supply of 9 V (Main), backed up by a permanent 6 V battery.

Serial Bit Rate

Baud rate	19200 baud
Start bit	1
Data bits	8
Parity bits	0
Stop bits	1

External Connections

- USB – mini B cable for RL78 microcontroller
- USB – B for FT230XS USB – UART

Appendix C

Circuit Diagrams

List of figures

Figure 1: RL78 microcontroller pin setup....26
Figure 2: Buzzer....26
Figure 3: Push button System....27
Figure 4: LCD Display....27
Figure 5: USB-UART....28
Figure 6: Voltage Regulator (5V)....28
Figure 7: Voltage Regulator (3.3V)....29
Figure 8: TEMPERATURE/Pressure/Humidity sensor....29
Figure 9: Wind sensor....30
Figure 10: Power Supply Sensor....31

RL78 Microcontroller Pin Setup

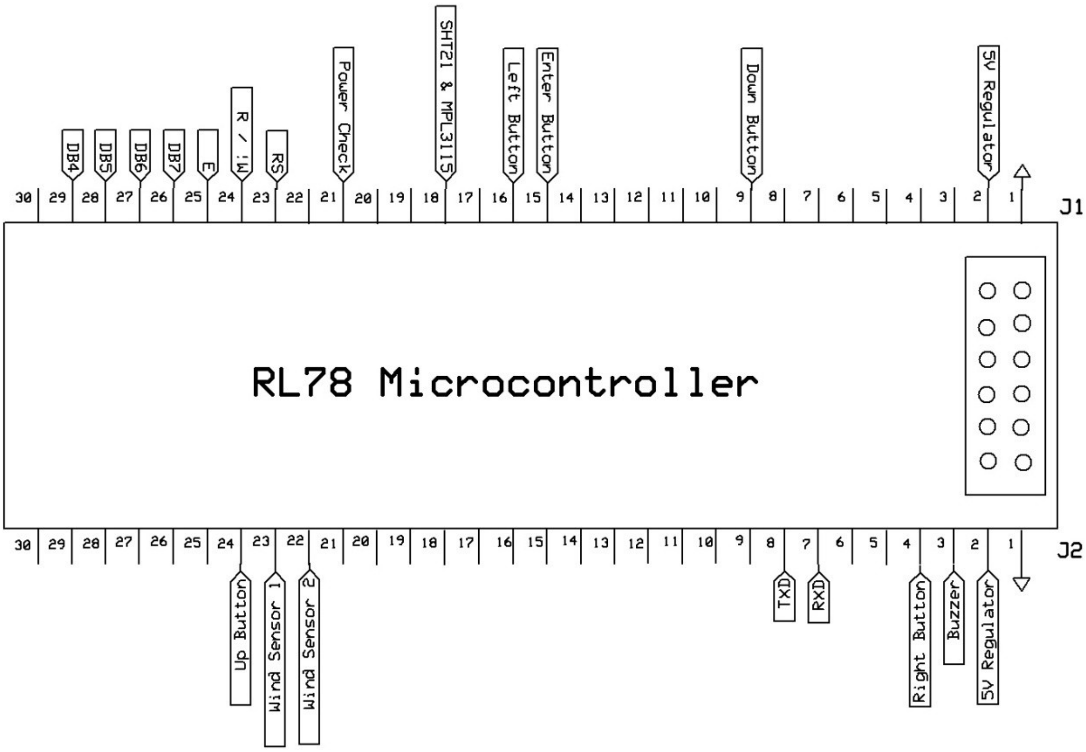


FIGURE 1: RL78 MICROCONTROLLER PIN SETUP

Buzzer

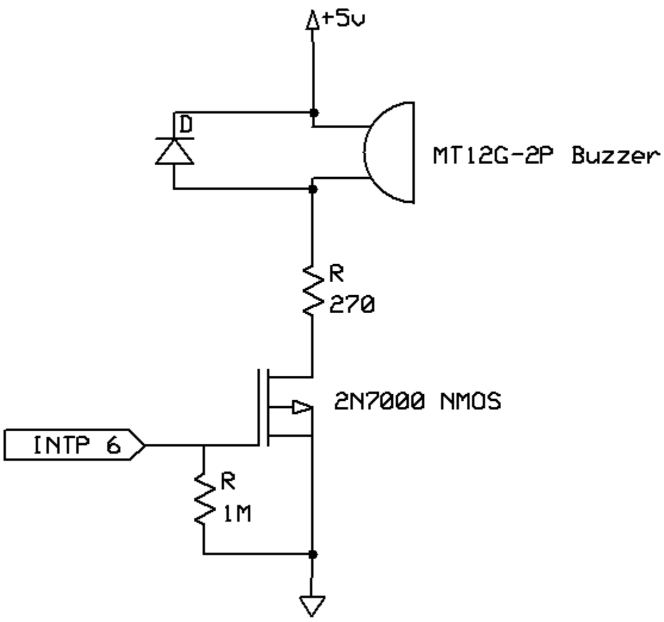


FIGURE 2: BUZZER

Push Button System

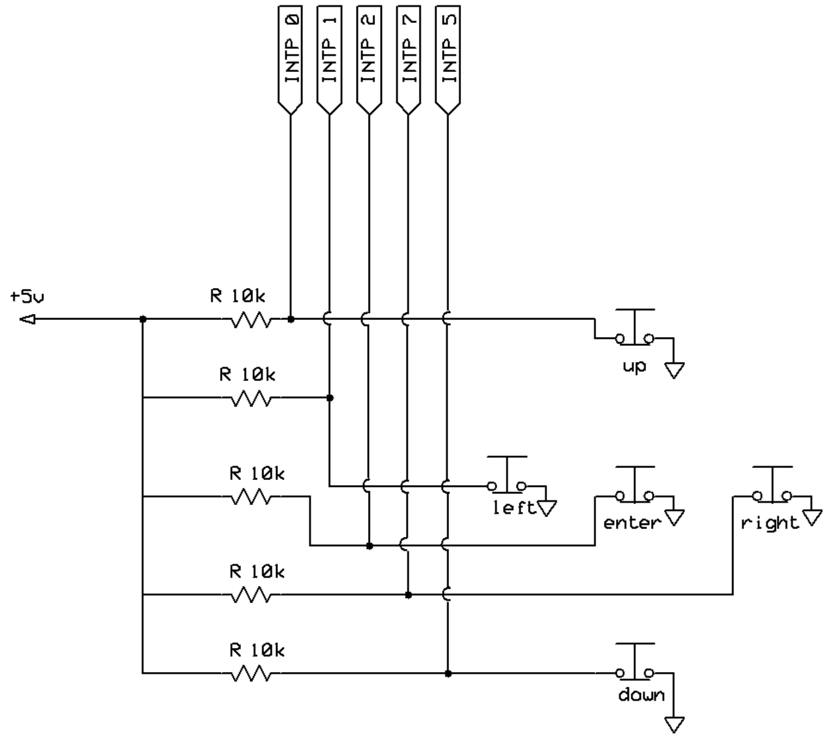


FIGURE 3: PUSH BUTTON SYSTEM

LCD Display

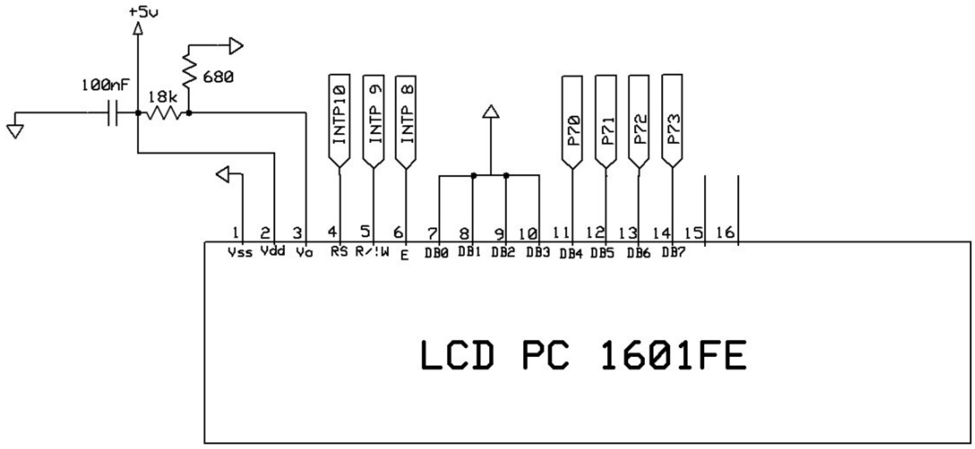


FIGURE 4: LCD DISPLAY

USB – UART

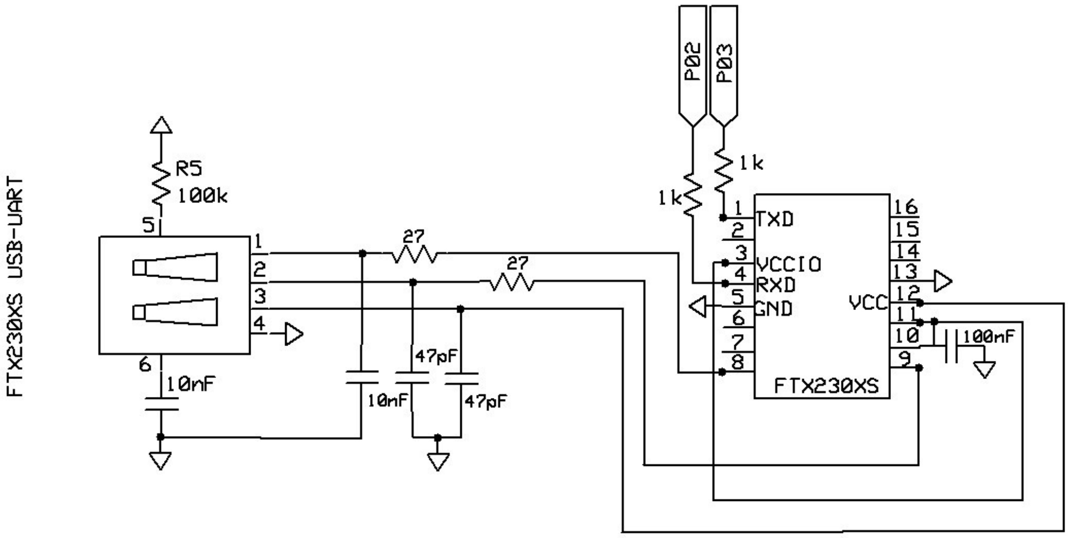


FIGURE 5: USB-UART

Voltage Regulator (5V)

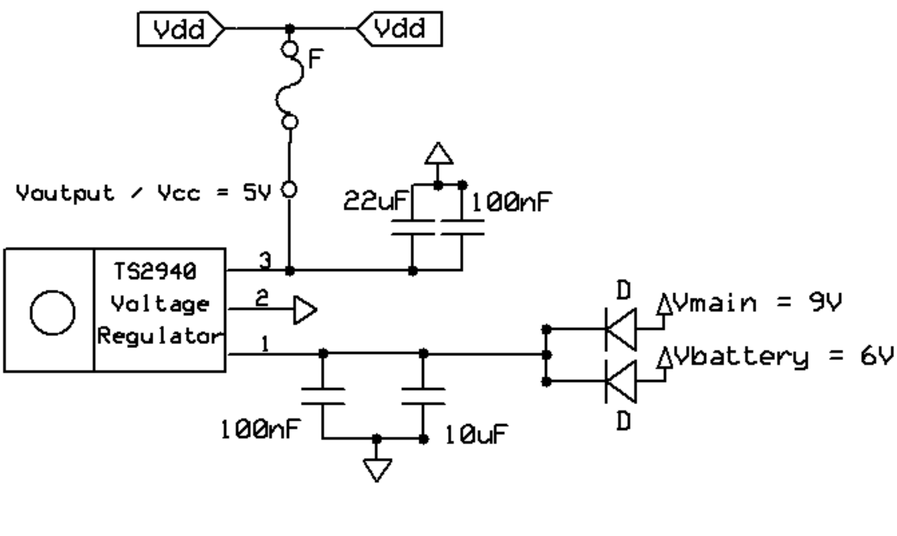


FIGURE 6: VOLTAGE REGULATOR (5V)

Voltage Regulator (3.3V)

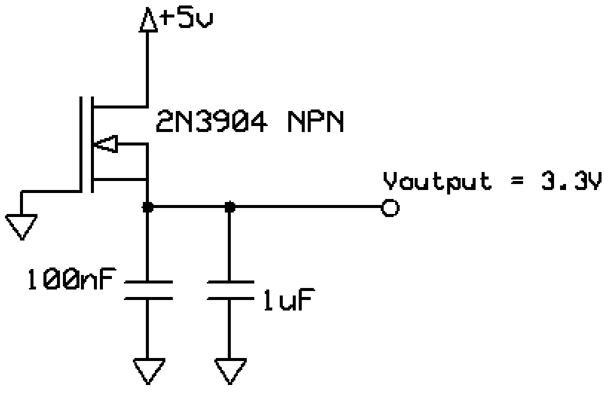


FIGURE 7: VOLTAGE REGULATOR (3.3V)

Temperature/Pressure/Humidity Sensor

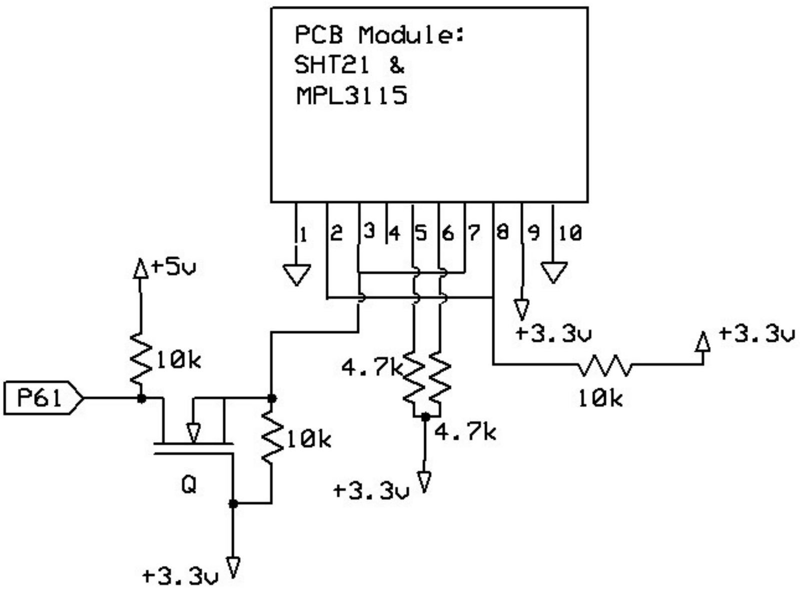


FIGURE 8: TEMPERATURE/PRESSURE/HUMIDITY SENSOR

Wind Sensor

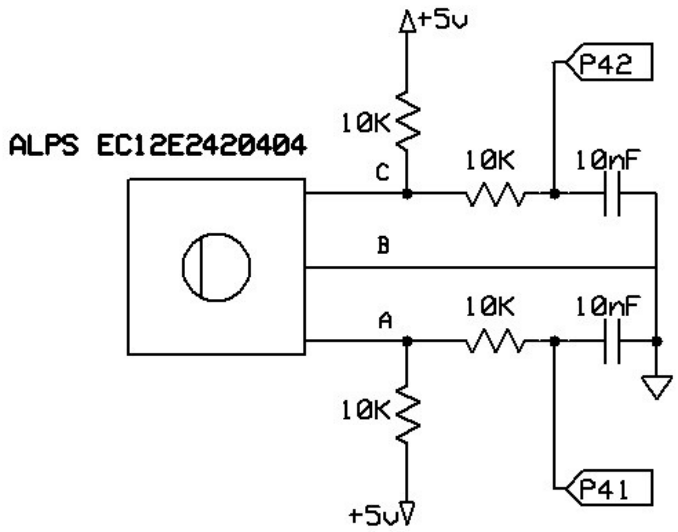


FIGURE 9: WIND SENSOR

Power Supply Sensor

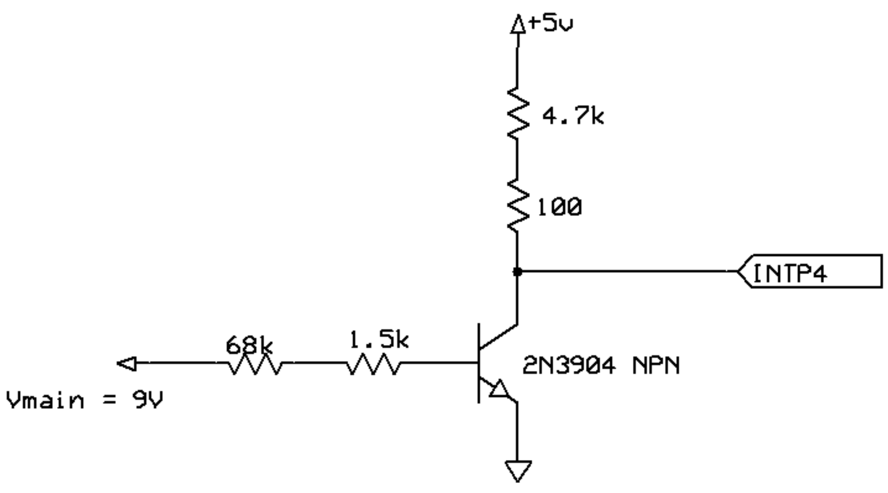


FIGURE 10: POWER SUPPLY SENSOR