12/10/2016

# DC Motor control using GSM module and other sensors

Final Project for ECEN 5613, Embedded Systems Design

Tarun Singh, Subhradeep Dutta
UNIVERSITY OF COLORADO BOULDER

# Contents

1. **INTRODUCTION** ........ **2**
2. **SYSTEM OVERVIEW** ........ **2**
3. **TECHNICAL DESCRIPTION** ........ **3**
   A. BOARD DESIGN ........ 3
   B. PIR SENSOR ........ 3
   C. GSM MODULE ........ 5
   D. ACCELEROMETER ........ 6
   E. FLEX SENSOR ........ 7
   F. ANALOG TO DIGITAL CONVERTER ........ 7
   G. MOTOR DRIVER ........ 9
   H. FIRMWARE DESIGN ........ 10
   I. TESTING PROCESS ........ 12
4. **RESULTS AND ERROR ANALYSIS** ........ **13**
5. **CONCLUSION** ........ **14**
6. **FUTURE DEVELOPMENT IDEAS** ........ **14**
7. **ACKNOWLEDGEMENTS** ........ **14**
8. **REFERENCES** ........ **14**

## 1. INTRODUCTION

DC motors have varied applications in robotics, home automation, industrial automation, drones and literally anything that involves movement. Controlling the speed and direction of movement of DC motors is an essential part in building embedded systems that have the capability to move. Nowadays every electronic component around us is controlled by a microcontroller and hence the interfacing of a motor to a microcontroller allows us to control real world components using digital logic.

The 8051 microcontroller along with the STMicroelectronics L293D motor driver IC provides complete control over the speed and direction of the DC motor. Pulse width modulation is fed to the motor driver and by varying the duty cycle of the PWM (pulse width modulation), varying speeds are achieved.

## 2. SYSTEM OVERVIEW

Home automation is gaining increasing importance with the launch of products like Nest WiFi enabled thermostats. Working along the same lines, we have added GSM (Global System for mobile communication) connectivity to the motor thus empowering the user to vary speed and change the direction of motion for the DC motor. The PIR (passive infrared sensor) can be placed along the entrance of the house thus turning on the motor when the user enters and turning it off when the user exits the house.

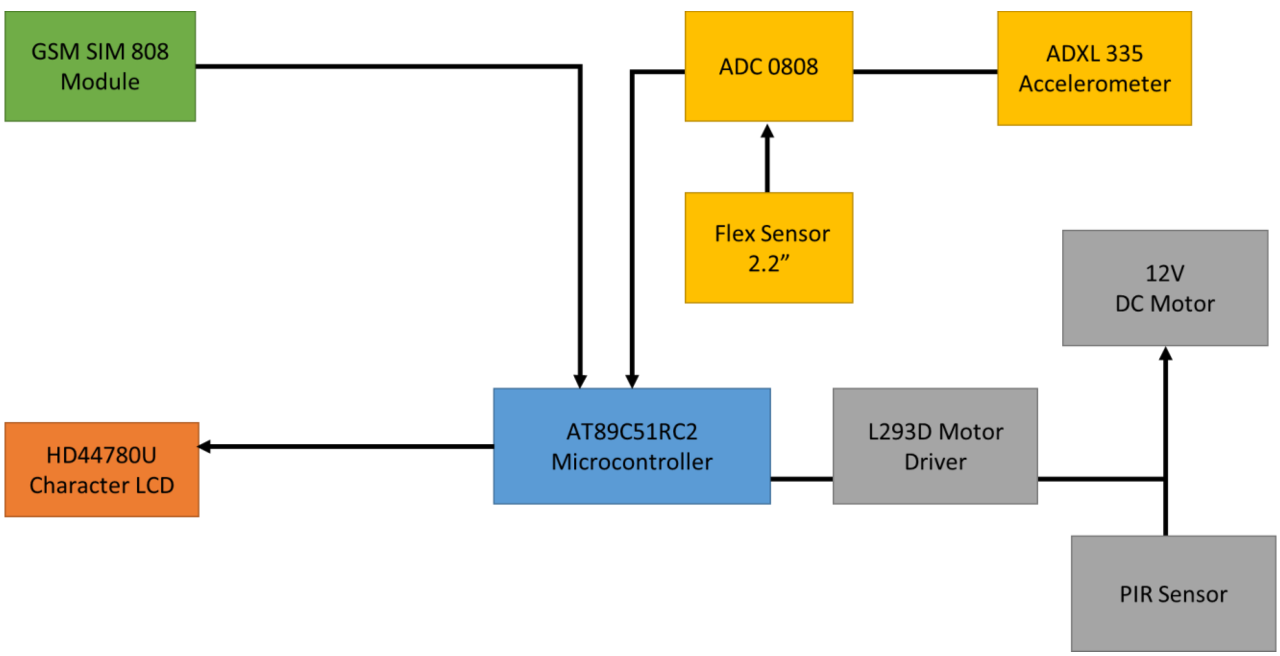


Figure 1: BLOCK DIAGRAM

Figure 1 represents the overall block diagram of the entire system. The GSM module can receive messages from the user and based on the content of the message the motor's speed or direction is controlled. The flex sensor and the accelerometer are also used to vary the speed of the DC motor. The PIR sensor behaves as a switch to turn on and turn off the motor. The LCD displays a graphical representation of the current duty cycle of the GSM module and also displays the duty cycle in numbers.

## 3. TECHNICAL DESCRIPTION

### a. BOARD DESIGN

The board uses the Atmel AT89C51RC2 microcontroller which has been used in Lab 3 and Lab 4 for ECEN 5613. The GSM module's RX and TX pins are connected to the TX and RX pins of the 8051 microcontroller thus allowing communication through UART. The GSM module uses AT command standards while communicating through UART. The power supply and GND was supplied from a USB port using a USB to TTL converter cable.

The L293D motor driver is used to drive the 12V DC motor. A DC voltage supply was used to provide the 12V supply to the motor driver which was used to control the movement of the motor. The GPIO pins P1.0 and P1.2 was used as motor inputs for the driver.

The PIR sensor was powered using a 5V supply from the board and GND respectively. The PIR sensor senses a change in infrared radiation and triggers a logic output high on its output pin. This output is directed to a SPLD which negates the logic high output to logic low output. The SPLD output is then routed to the INT0 external hardware interrupt pin of the 8051.

The flex sensor was connected along with a voltage divider circuit and varying amounts of voltage was calibrated to varying duty cycle for the PWM.

The accelerometer was also powered using the board 5V supply and its output voltage from the Z axis was mapped to varying duty cycles for motor speed control.

The ADC was used to convert the output of the flex sensor and the accelerometer to its digital equivalent.

### b. PIR SENSOR

Figure 2: Adafruit PIR sensor

The Adafruit PIR motion sensor is capable of sensing movement due to the change in the amount of infrared radiation incident on it. The sensor is split into two halves. The outputs from each of these halves cancel each other out. If one half senses more infrared radiation as compared to the other half then the output goes high. The PIR sensor employs a BISS0001 motion detector IC which converts the analog output of the PIR sensor to a digital output. The PIR sensor also uses a NIR11NH Fresnel lens setup to focus the incoming infrared radiation to the center of the sensor setup for more accurate results. The actual sensor is a RE 200B pyroelectric sensor.

The Adafruit PIR sensor comes with built-in sensitivity adjustment which was initially tweaked to match the sensitivity needs for our project. There is also an adjustable jumper on the sensor which enables or disables retriggering. Usually the output of a PIR sensor keeps switching between high and low as long as the sensor senses a differential infrared radiation. But this kind of output is not suitable for most purposes. So the retriggering option comes to play here. When retriggering is enabled, the sensor output is high for the entire time as long as the sensor senses movement. This kind of output is exactly what we need to generate an interrupt signal for the 8051 microcontroller since we are using edge sensitive interrupts.

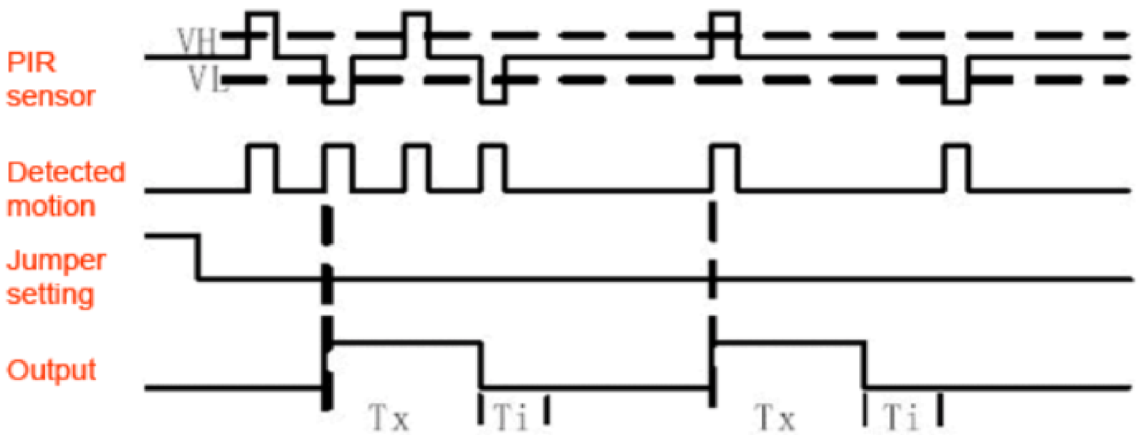


Figure 3: PIR sensor output without retriggering

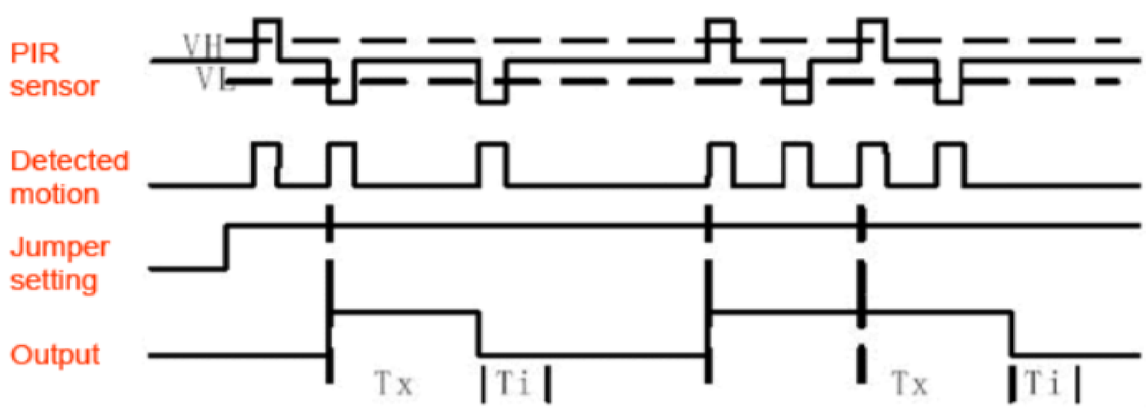


Figure 4: PIR sensor output with retriggering

There are two timeouts associated with the PIR sensor namely:

Tx: This indicates how long the output is maintained high after the sensor detects movement. This can be adjusted by using the built in adjustment knob.

Ti: This indicates how long the output is guaranteed to be off when there is no movement detected.

| PIR Sensor | Connected to |
|---|---|
| Vcc | Vcc of 8051 board |
| GND | GND of 8051 board |
| Digital Out | Input pin of SPLD |

**c. GSM MODULE**


Figure 5: Adafruit SIM808 GSM module

The Adafruit SIM808 GSM module supports quad band GSM connectivity and includes built-in GPS capability. It also has TTL logic pins thus avoiding the need for a MAX232. The module also needs a separate uFL GSM antenna which has to be purchased separately and has the option of being powered by a lithium-ion battery. A GSM sim is required to be attached into the module. Some of the important pins and their functions are described below.

- Vio – A 5V supply needs to be provided to this pin for the module to work
- Key – This pin is used to turn the module on or off.
- 5V – This pin can be used to recharge the battery
- PS – This is the power status pin which indicates when the module is turned on or turned off.
- NS – This pin indicates the current network status
- Reset – This pin can be toggled low for at least 100ms to generate a hard reset.

- RX & TX – These are the UART pins used for communicating with the microcontroller.

### d. ACCELEROMETER

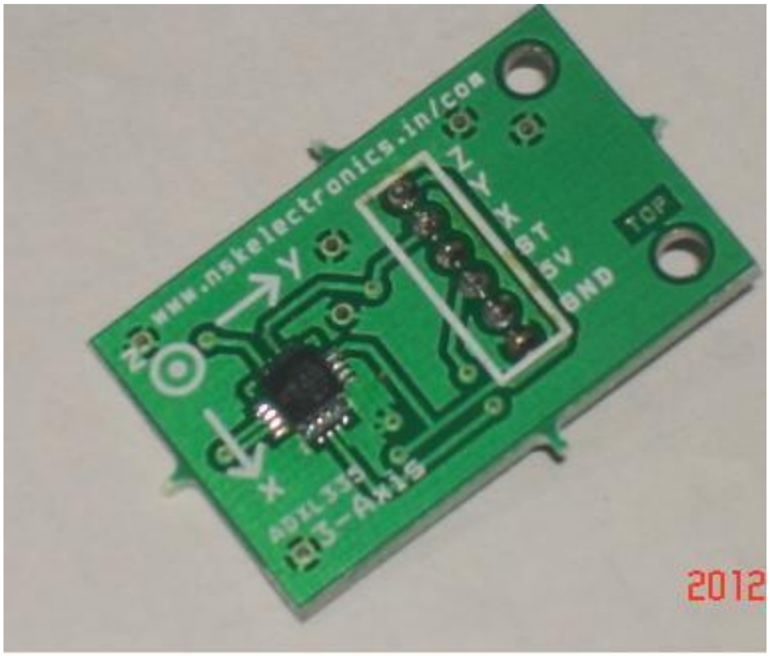


Figure 6: ADXL335 accelerometer

We have used the ADXL335 3-axis accelerometer for this project. The accelerometer is used as a speed regulator for the DC motor and the output voltage from the Z-axis pin of the accelerometer is mapped to varying duty cycles for motor control.

The ADXL335 is a three axis accelerometer with a maximum range of -3g/+3g. The output voltage of the ADXl335 is ratiometric, i.e., for a Vs=3V the output at 0g is Vs/2 which is 1.5V. This accelerometer's bandwidth can be adjusted using the Cx, Cy and Cz capacitors connected to Xout, Yout and Zout. The noise for the accelerometer is not ratiometric hence a higher supply voltage results in lower noise on the output.

| **Accelerometer** | **Connected to** |
|---|---|
| Vcc | Vcc of 8051 board |
| GND | GND of 8051 board |
| Zout | Input Channel 1 of ADC |

### e. FLEX SENSOR


Figure 7: Flex Sensor

The Spectra Symbol flex sensor behaves like a variable resistor when bent over a certain angle and can be used to convert displacement along an angle into electrical output. The resistance across its two ends is proportional to the angle at which the sensor is bent. The sensor has a resistance range of 40 KΩ to 220 KΩ. This sensor along with an appropriate voltage divider circuit was used to resemble the speed regulator for the DC motor.

| Flex Sensor | Connected to |
|---|---|
| Terminal 1 | Vout of voltage divider circuit |
| Terminal 2 | GND of 8051 board |

### f. ANALOG TO DIGITAL CONVERTER

Figure 8: Analog to Digital converter

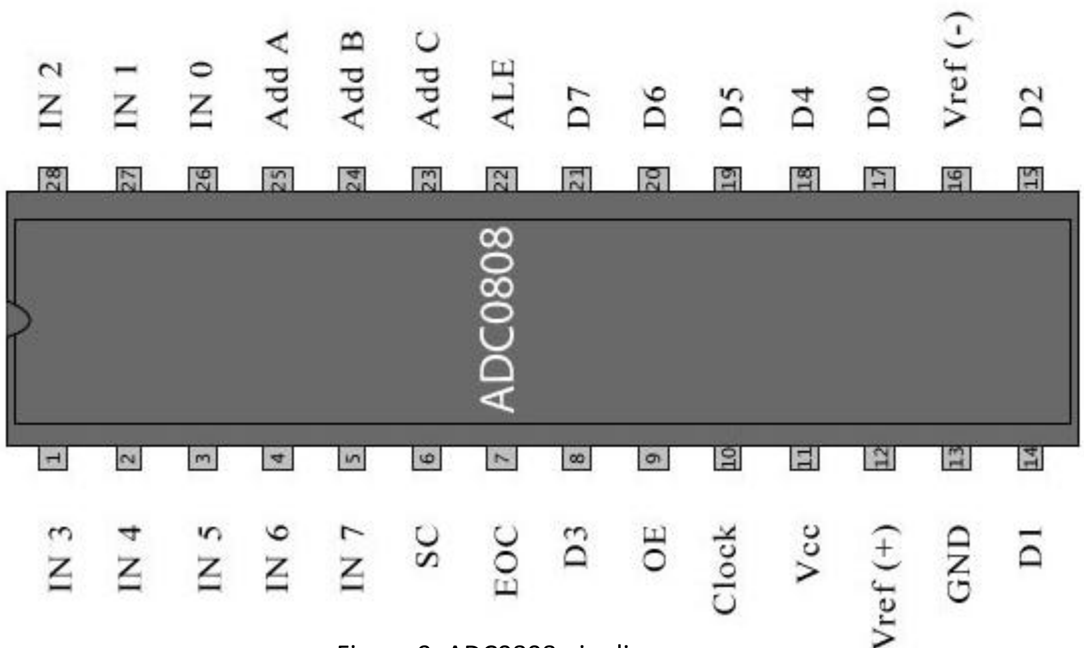


Figure 9: ADC0808 pin diagram

Since the output of the accelerometer and the flex sensor is analog voltage so they must be converted to their corresponding digital equivalents for the microcontroller to operate on their outputs. The ADC0808 from National Semiconductor has been used for this purpose. ADC0808 is an 8 channel 8 bit resolution ADC which relies upon the method of successive approximation to convert the analog signal to digital output. The ADC requires the following connections to successfully convert an analog input to its corresponding digital output.

- The IN0-IN7 pins receive the analog input signal. There are total 8 different channels and any one of them can be selected using a built in multiplexer. IN0 is connected to the flex sensor output and IN1 is connected to the accelerometer output.
- Address pins A, B, C are used to select one of the input channels from IN0 to IN7. Address pin A refers to the LSB and address pin C refers to the LSB.
- SC and EOC refer to start conversion and end of conversion respectively. The SC receives a low to high pulse indicating the ADC to start conversion and EOC goes from low to high indicating that the conversion is complete.
- The ALE is used to latch the address on to the ADC so that even if there are any changes in the address pins the ADC would ignore that.
- The Vref(+) and Vref(-) correspond to the maximum reference positive voltage and maximum reference negative voltage based on the which the ADC determines the digital output. The difference between Vref(+) and Vref(-) is divided into 255 equal divisions and this is referred to as the minimum step size. The minimum step size in our project is 20mV.

- A high to low pulse to the OE pin is used to read the data from the data pins of the ADC.
- A clock pulse is given to the Clock pin of the ADC. A wide range of frequencies from 10 KHz to 1280 KHz is supported by this ADC.

| ADC | Connected to |
|---|---|
| IN0 | Vout of voltage divider circuit |
| IN1 | Zout of accelerometer |
| SC | P1.0 of 8051 |
| EOC | P1.4 of 8051 |
| Clock | P1.6 of 8051 |
| Vcc | Vcc of 8051 board |
| GND | GND of 8051 board |
| Vref(+) | Vcc of 8051 board |
| Vref(-) | GND of 8051 board |
| OE | P1.5 of 8051 |
| ALE | P1.7 of 8051 |
| D0-D7 | D0-D7 of 8051 |

### g. MOTOR DRIVER


Figure 10: H bridge motor driver

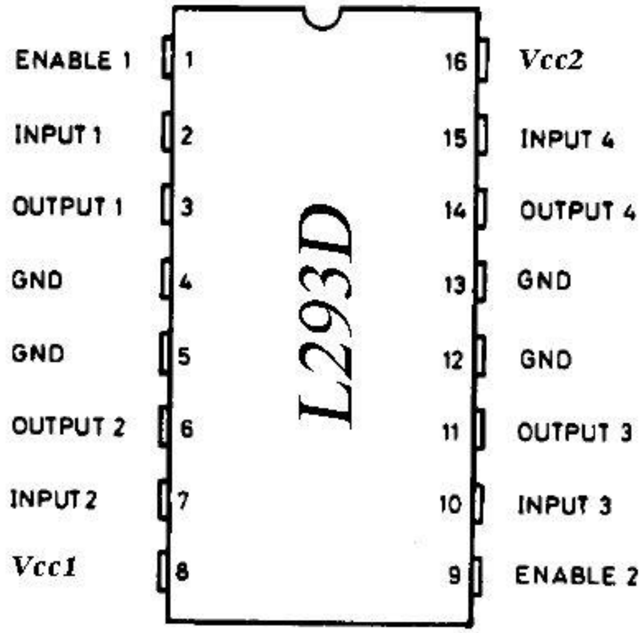


Figure 11: H bridge motor driver pin diagram

The DC motor needs a 12V supply to drive it. To provide the 12V supply to the motor a motor driver needs to be used. The L293D has two full H bridge circuits that can be utilized to drive two different DC motors.

| Microcontroller | L293D Driver |
|---|---|
| P1.0 | Input 1 |
| P1.2 | Input 2 |

- The enable pin is fed with the PWM from the microcontroller.
- The two inputs are provided from the microcontroller pins P1.0 and P1.2. They can be used to control the direction of motor’s rotation as shown below.

| Microcontroller Pins | | Direction Control |
|---|---|---|
| P1.0 | P1.2 | |
| 0 | 1 | Counter clockwise |
| 1 | 0 | Clockwise |

- Vcc1 is fed with the 12V DC supply from an external source.
- Output1 and Output2 pins of the drivers are connected to the two terminals of the DC motor.

### h. FIRMWARE DESIGN

Firmware design for this project included writing code for the GSM module, the PIR sensor, the motor driver and for the ADC. The firmware was developed using SDCC 2.6.0 along with the CodeBlocks 12.11 IDE for code development. SDCC’s large memory model was used.

- **MOTOR DRIVER**

The motor driver (motor.c) defines the functions to control the speed of motor as well as the direction. All the functions were written in C. The detailed code with comments can be found in the appendix.

| Function Name | Description |
| --- | --- |
| motor_driver | Reads the message from GSM module to change speed or direction |
| display_duty_cycle_ones | This function displays the ones digit of the duty cycle on the LCD |
| display_duty_cycle_tenth | This function displays the tens digit of the duty cycle on the LCD |
| slow_down_motor | Decreases duty cycle by 5% until the desired duty cycle is achieved |
| speed_up_motor | Increases duty cycle by 5% until the desired duty cycle is achieved |
| motor_on | Used to turn the motor on at default 50% duty cycle |
| motor_off | Used to turn the motor off |

- **GSM MODULE**

For the GSM module, we are polling within the main program (main.c) for an incoming SMS message and based on the contents of the SMS, the necessary action is taken. If the message has commands for speed control or change of direction, then the appropriate functions are called else the incoming SMS message is displayed on the LCD.

- **PIR SENSOR**

The PIR sensor acts as a switch and turns on or turns off the motor whenever it is triggered by a change in incident radiation. So the output of the PIR sensor through the SPLD triggers an external hardware interrupt. Once the control enters the ISR, the motor is turned on or off depending on the previous state of the motor.

| Function Name | Description |
| --- | --- |
| external0 | Turns the motor on/off whenever the PIR sensor is triggered |

- **ANALOG TO DIGITAL CONVERTER**

The ADC is used within the timer interrupt ISR to read data from the flex sensor as well as the accelerometer to modify the motor’s duty cycle.

| Function Name | Description |
| --- | --- |
| timer0interrupt | Changes duty cycle by reading data from the ADC for both the flex and the accelerometer |

### i. TESTING PROCESS

For the purpose of testing the entire setup, each component was first setup as a separate entity and after it was fully functional it was integrated into another board with the rest of the working components combined together. The entire process is shown below.

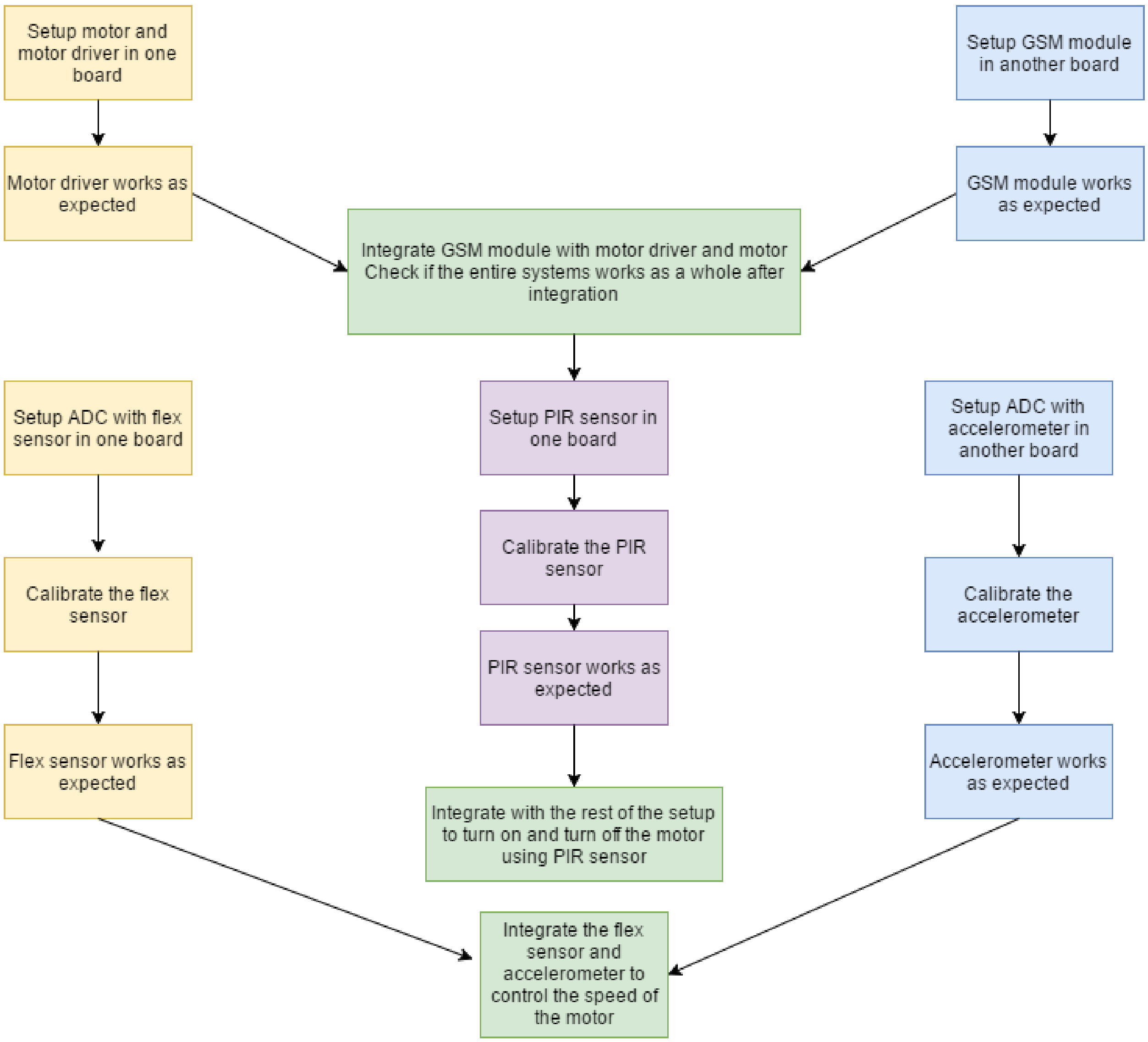


Figure 12: Testing process

Once all the components were integrated, the system was tested by sending text messages at different periods and a change in the motor duty cycle along with the corresponding output on the LCD was observed.

## 4. RESULTS AND ERROR ANALYSIS

The system worked together as a whole since we tested each component individually and then integrated it with the rest of the setup. But we ran into some issues trying to control the motor using all the different sensor inputs.

### a. LACK OF INTERRUPT SOURCE FOR THE ADC

The ADC was used to convert the flex sensor and accelerometer output but there was no signal that could indicate when to start conversion using the ADC. Hence we decided to use the timer interrupt that would read the values from the ADC for both the flex sensor as well as the accelerometer every alternate second. This also caused issues because now we were trying to control the same thing through two different sources at the same time resulting in a conflict. Every time the values were read from the ADC, the duty cycle was set as per the flex sensor and then the duty cycle was set as per the accelerometer. This resulted in abrupt changes in the motor duty cycle due to data being read from both the sensors simultaneously. Hence we decided to move the flex sensor and accelerometer setup to a different board. And also added a switch that would toggle control between the flex sensor and the accelerometer. This way there was no conflict between the two sensors trying to control the same entity at once.

Another way to address this issue would be to setup a switch that would toggle control between the GSM module, the accelerometer and the flex sensor so that only one of these was functional at a time. This would avoid conflict between the three modules trying to control the motor all at the same time.

### b. PIR OUTPUT SIGNAL IS ACTIVE HIGH

Initially the PIR sensor output was connected to the interrupt pin of the microcontroller directly because we assumed that the falling edge of the output from the PIR sensor would trigger the interrupt. But the results after the connection were not as expected and they were delayed due to significant time delays that are needed each time after the PIR sensor senses movement and generates an output. Then we decided to route the output of the PIR sensor through the SPLD and negate it at the output from the SPLD so that as soon as the PIR sensor triggered movement, immediately the motor was turned on or off.

### c. PIR SENSOR TOO SENSITIVE

The initial idea was to turn on and off the motor by swiping the hand over the PIR sensor. But the PIR sensor was so sensitive that even movement at the far end of the lab triggered the PIR sensor. To overcome this issue we 3D printed an enclosure that would prevent stray radiations from setting off the PIR sensor. After adding the enclosure the results were extremely favorable. The PIR sensor now responded only when we swiped our hand over the sensor.

### d. VERBOSE GSM MODULE

The GSM module was extremely verbose in terms of its output so a lot of filtering was required to fetch the actual SMS content and ignore the other data coming along with it.

### e. MOTOR ON OFF

We initially assumed that turning off the PWM would turn off the motor but this was not the case. The motor inputs to the driver also needed to be changed to either zeroes or ones to turn off the motor.

## 5. CONCLUSION

The entire project was a great learning experience for both of us since this was our first attempt at an embedded system project. We were satisfied with the amount of work we were able to get done for the project and the final outcome. It gave us a good foundation on motor control and will pave the path for future projects that involve more movement using motors such as moving robots or drones. There was sufficient amount of hardware and software involved to make the project challenging as well as interesting at the same time.

One thing that we realized was that the idea of trying to control one particular element from so many different inputs was bound to cause conflicts. The entire project design was focused on trying to control the DC motor speed based on inputs from the ADC, PIR sensor and the GSM module. It would have been a better idea to include a switch to toggle between these elements or include some other components that could be controlled by the ADC and PIR sensor while the GSM module would be solely in charge of motor control.

## 6. FUTURE DEVELOPMENT IDEAS

One idea that we had in mind but could not implement due to lack of time was motor control using gestures. The ADPS 9960 is a RGB and gesture sensor that recognizes various hand movements. It has an I2C interface which can be connected to the 8051. Once that is achieved, different gestures can be used to turn the motor on/off or vary the speed and direction of the motor. This would indeed turn out to be an interesting project to develop.

Apart from a motor, a few other components can also be combined such as a light setup, few relays and a heating control system (thermostat) so as to resemble a complete home automation system.

## 7. ACKNOWLEDGEMENTS

We would like to thank the instructor, Professor Linden McClure for this challenging course and the wonderful opportunity to work on such a challenging project.

We would also like to thank all our fellow classmates who gave their useful inputs, suggestions and debugging advice to further improve this project.

We would like to acknowledge and thank the various sources listed below.

## 8. REFERENCES

i. Figure 2, 3, 4 from https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf
ii. https://cdn-learn.adafruit.com/downloads/pdf/adafruit-fona-mini-gsm-gprs-cellular-phone-module.pdf
iii. https://cdn-learn.adafruit.com/downloads/pdf/adafruit-fona-3g-cellular-gps-breakout.pdf
iv. Figure 5 from https://cdn-learn.adafruit.com/downloads/pdf/adafruit-fona-808-cellular-plus-gps-breakout.pdf
v. Figure 6 from http://www.nskelectronics.com/adxl_335.html
vi. Figure 7 from https://solarbotics.com/product/50808/
vii. Figure 8 from http://www.dnatechindia.com/ADC0808.html

viii. Figure 9 from http://www.engineersgarage.com/electronic-components/adc0808-datasheet
ix. Figure 10, 11 from http://blog.jayroboticsclub.in/how-ic-l293d-runs-a-motor/
x. http://datasheet.octopart.com/L293D-STMicroelectronics-datasheet-10011.pdf
xi. www.learn-c.com/adc0809.pdf
xii. https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/FLEX%20SENSOR%20DATA%20SHEET%202014.pdf
xiii. https://www.sparkfun.com/datasheets/Components/SMD/adxl335.pdf