# Mini Smart Home

MICROPROCESSOR COURSE PROJECT

Sushma Sunkollu Nagaraj | Microprocessor Systems | 8th May,2019

**OBJECTIVE**

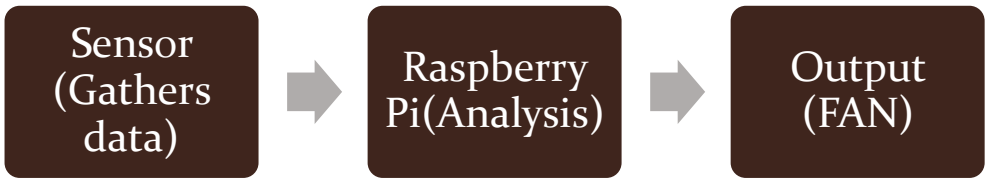
The main objective of the project is to design a home automation system. The design is just a prototype of a smart system. However, several other sensors and components can be integrated with the design to make it more sophisticated. The project also helps to get a better understanding of programming with Raspberry Pi, C coding, analog circuits, Interrupts and Polling.

**INTRODUCTION**

**"Home Automation"** is something that is trending today. In this project an attempt has been made to design certain modules of a smart home system. The two modules are:

a) Temperature Controlled Fan (motor) using Raspberry Pi and DHT11 sensor to record temperature, humidity and control the speed of the fan to manage heat. For each measurement cycle the sensor gathers the data and transmits the same serially to Raspberry Pi for processing. The code will then analyses data and then drives the fan(motor) accordingly using the PWM.
b) Integration of Motion Sensor like a PIR Sensor with Raspberry Pi which turns on an LED/triggering the buzzer when motion is detected.

**SYSTEM DESIGN, COMPONENTS USED AND BLOCK DIAGRAM**




**Module-1**

Module-1 is the first part of the design. In this module the speed of the fan is controlled using the PWM signal based on the input from the sensor. In this module, I have used a DHT sensor. DHT11 sensor consists of two components, the humidity sensing component and temperature sensor(thermistor) and an IC at the back of the sensor. The humidity sensing component has two electrodes with moisture holding layer in between the two layers and when there is change in humidity the conductivity between the two electrodes also changes which is detected by the IC and the sent to controller. The thermistor is a variable resistor which changes when there is change in temperature. The resistance decreases with increase in temperature.

**Integration of the module with the design:** The DHT11 sensor is a 3-pin sensor with Vcc, GND and the third goes to Raspberry Pi. In order to initiate the process, the DHT11 sensor must be reset and this can be done by assigning a zero to the output. In order to set it to the output mode so that it can be invoked to send the data we need to write a low and high to the sensor for about 18 and 40 microseconds respectively. In this case polling is being used and we keep checking if the sensor has completed obtaining the required information for further processing thus waiting for maximum time required by the sensor to capture required amount of information. The information captured will be serially transmitted to raspberry pi, once done, we then check if the data is valid using the checksum function. If the data is valid then we begin further processing. The sensor output is in the form of an array of size 5. The first two indexes of the array indicate the humidity (whole and decimal part) and next two indexes indicate the temperature information (whole and decimal part) and the last indicates the value of checksum. Based on the temperature obtained we can set the duty cycle (PWM). According to the specifications related to the sensor, the duty cycle in this case is not in terms of percentage but it is in the form of whole number. A 100% duty cycle corresponds to a value of 999 and a duty cycle of 60% corresponds to value of 614 and so on. Based on the temperature obtained using the pwmWrite pin we control the speed of the motor.

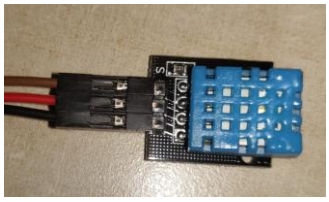


**Figure: DHT11**

Motor Driver Circuit: In order to control the speed of the motor using PWM I have built a motor driver circuitry using the L293d Motor driver IC. It acts as an interface between my Raspberry Pi and motor. The most important advantage of this circuit is that it allows the motor rotation in both forward and reverse direction. This is possible due to the H-Bridge within the IC.

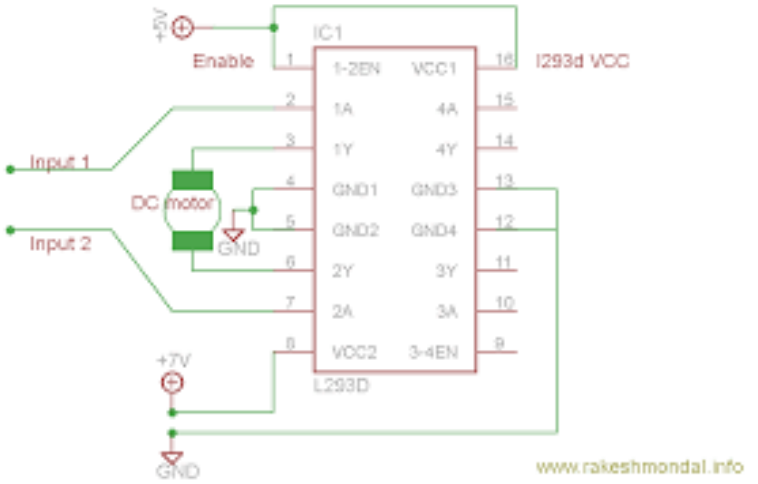



**Figure: L293D**

**Modue-2:** Integration of the PIR Sensor with Raspberry Pi for motion detection

When the motion is detected the PIR sensor, the analog input obtained is converted to digital output to drive the LED. The ADC in this case has been implemented using the LM339 IC (Comparator). The pin 2 of the comparator is active low gives an inverted signal. Thus, to drive the LED I have used a pull up resistor of 1k and connected it to supply. The working of the comparator and waveforms are shown below:

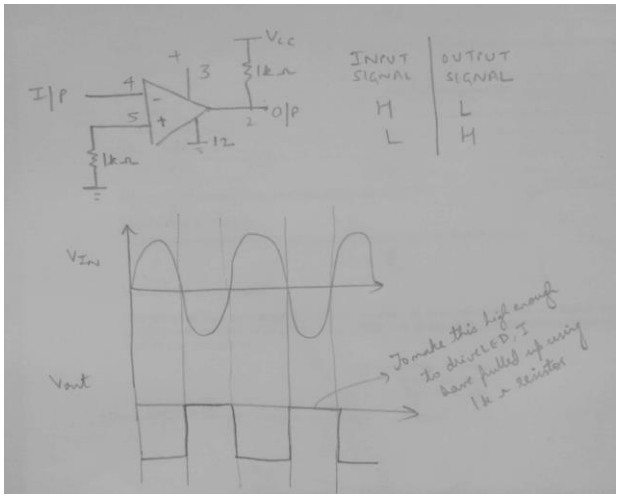



**Figure: ADC**

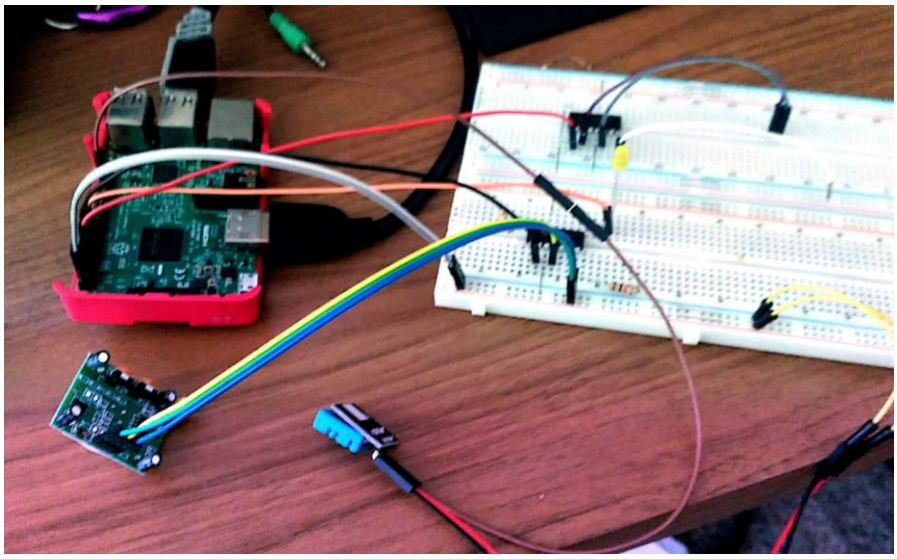


Figure: Set-Up

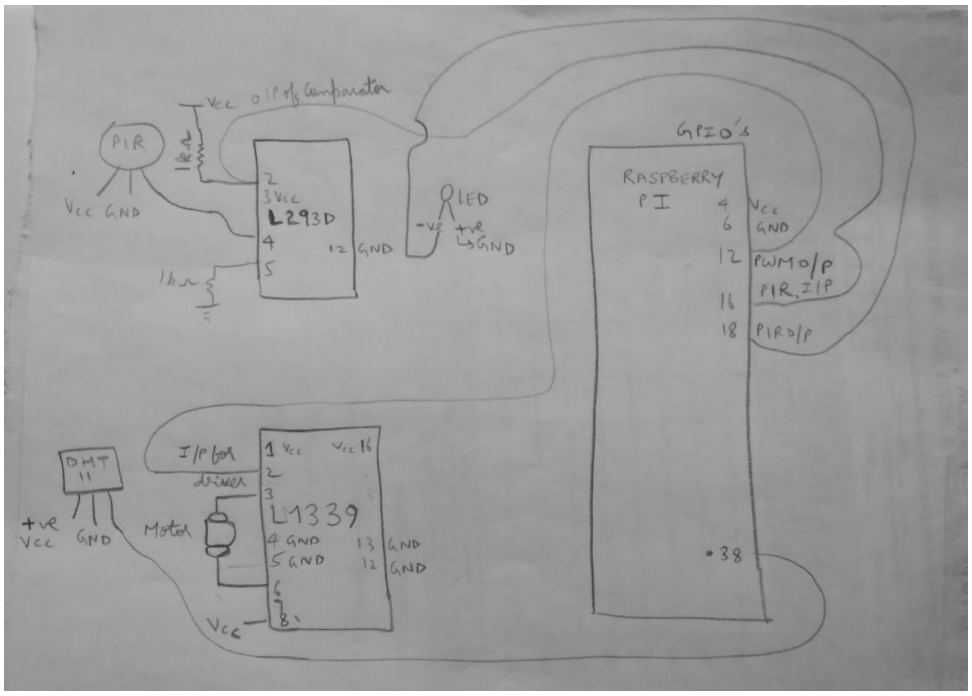



**Figure: Schematic of entire Circuit**

**CHALLENGES FACED/LESSONS LEARNED:**

- Understanding how to convert the input data from sensor and send it to Raspberry for processing (Building a ADC)
- Understanding the working of Comparator used in ADC built.
- Understanding the pattern of input obtained from the DHT11 sensor, configuring it and extracting the required information for processing.
- Implementing the polling and interrupt for the sensors.
- Understanding the Motor Driver circuit and pattern in which PWM should be used to control the speed of Motor.
- Working with and configuring Raspberry Pi.
- Learnt how to use the wiringPi.

**SCOPE FOR IMPROVEMENT**

- In this project the humidity information obtained has not been used. The humidity information obtained can be used to control a humidifier.
- The input from the PIR sensor is just used to trigger and turn on the LED when motion is detected using interrupt. The same could be further extended to control several other devices at home.
- Using a DHT22 instead of DHT11 can give more accurate results.
- The motor driver circuits needs a brushed motor and it requires a very high voltage of 24V to drive it. Thus, instead of using a LED to show variation/brushless motor a better capable motor could have been used.
- Voice command operation of device can also be an excellent addition to this project.