**SYSTEM MODULES**

**1.1:PLANT AUTOMATION & MONITORING**

**1.1.1:Description**

We all like to update our status on social networking sites like twitter, Facebook etc. about all the things happening in our lives. So what if a plant updates its status on Twitter while you are on your holidays!!! Yes we are going to implement such a talking plant.

In this module we are using a moisture sensor (name), HT121E (encoder), HT121D (decoder), RF transmitter and receiver which are all powered by 3.3V source. The RaspberryPi (RPi) handles the Twitter update and sending notification over the Internet. The watering of the plant when the moisture is low is handled by an Atemga8, L293D and watering pump which is powered by a 9V source.

The moisture sensor transmits a high when it detects that the soil moisture is below threshold which is encoded by HT121E and transmitted serially to the RF transmitter. The RF receiver picks up the signal and transmits it to the decoder which is connected to the RPi. The RPi detects the signal and accordingly updates the status of the plant over Twitter using Twitter application configured in it. Also it commands the Atmega8 to start the watering pump so that the plant is watered for 3-4 seconds after which the pump is stopped. Since motor requires large current which cannot be provided by Atmega8, L293D motor driver is used. The code is written such that if the plant is wet the Twitter update occurs only in intervals of 30mins, and when the plant becomes dry, the status is tweeted instantaneously. Twitter does not allow duplicate tweets over a period of 24 hrs. Hence we have included the current time in the tweet so that each tweet is unique.

**Algorithm**:

* Create variables for temperature and humidity
* Set threshold values for temperature and humidity sensor
* Monitor temperature and humidity of surrounding atmosphere using temperature and humidity sensors
* Take input from temperature and humidity sensor
* Compare it with threshold values

1. If value is more than threshold value

send signal to water pump to sprinkle water on plants

* 1. If value is less than threshold, keep on monitoring

Update the status of plant on internet

**BLOCK DIAGRAM**

Twitter/PhoneEe

TemperatureSensor

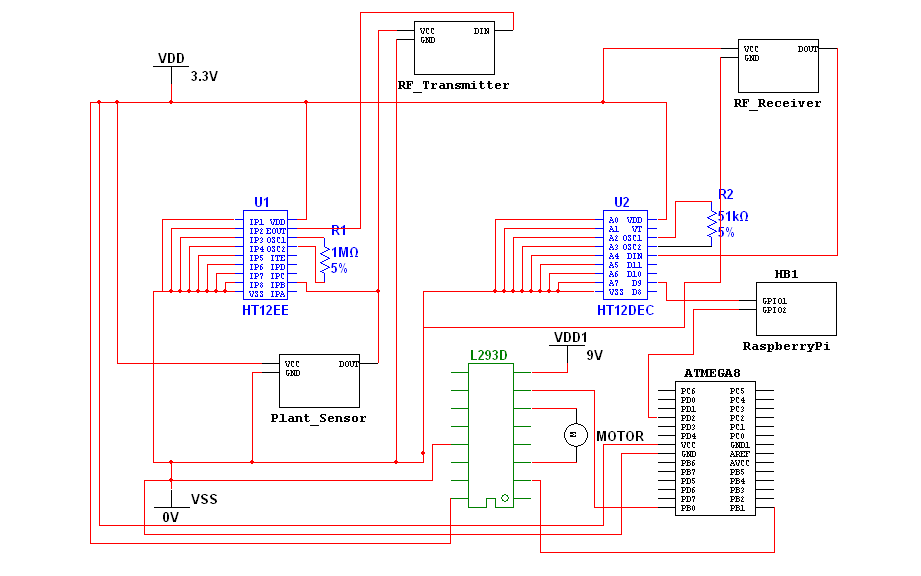
Update the status of the plant over the internet

Watering Unit

Raspberry PI

Plant

**1.1.2:CIRCUIT DIAGRAM**

**1.2:DOOR ALERT!**

**1.2.1:Description**

The door module consists of a magnetic switch, HT121E (encoder), HT121D(decoder), RF transmitter and receiver and RaspberryPi. The magnetic switch is connected to the door which will detect if it is closed or opened. If the door is opened the magnetic switch gets shorted and a high logic is passed to data pin of the encoder which transmits the signal serially to the RF transmitter. The RF receiver picks up the signal and gives it to the decoder which transmits the high signal to the RaspberryPI. The RPi sends a notification over the internet to your phone using Instapush android application that your door has been forcibly opened.

This module will help in preventing thefts while nobody is there at home or during the night when everybody is asleep. This module can also be used in jewellery shops, artifacts shops etc. Thus when the notification is received we can quickly alert our neighbors or police authorities. Also there will be a option of disabling the notification when not needed. This module is easy to implement and integrate with your phone and provides a cost effective solution for the near future where everything will be connected to the internet.

**Algorithm**:

* The Reed Switch is connected to the door
* If the Door is open, reed switch is closed and the RF module transmits the signal to the Raspberry-Pi
* Raspberry-Pi updates the status by sending a notification to phone through internet

**BLOCK DIAGRAM**

Your door has been opened

Raspberry PI

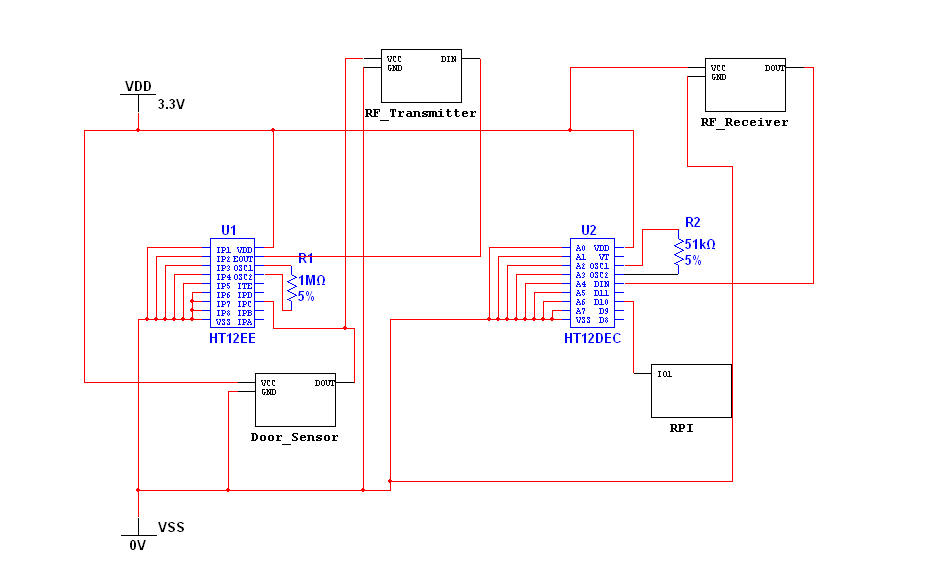
Radio Frequency Transmission

DOOR

SENSOR

Sends notification over the internet to your phone

**1.2.2:CIRCUIT DIAGRAM**

****

**1.3:SMART TRASH**

**1.3.1:Description**

You may have seen those overloaded trash cans in parks and crap spread around it. Not just they spoil the beauty of the place but also are the sources of many diseases. How about a trash can which can dispatch waste in junk yard whenever they get loaded fully?

Our smart trash does the same!

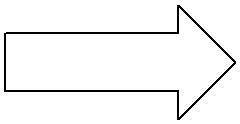
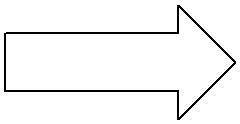
It consists of a line following bot over which the trash can is placed, a mechanical arm which will lift the trash can, some switches and a proximity sensor. A switch connected at the top of can gets pressed when trash can is full and sends a signal to the line following bot which drives the trash can to a main trash can placed nearby. A mechanical arm situated at the main trash can will lift the trash can and dump the trash in the former. A proximity sensor connected to Atmega16 of the line following bot detects the main trash can after which the waste is dumped.

**Algorithm:**

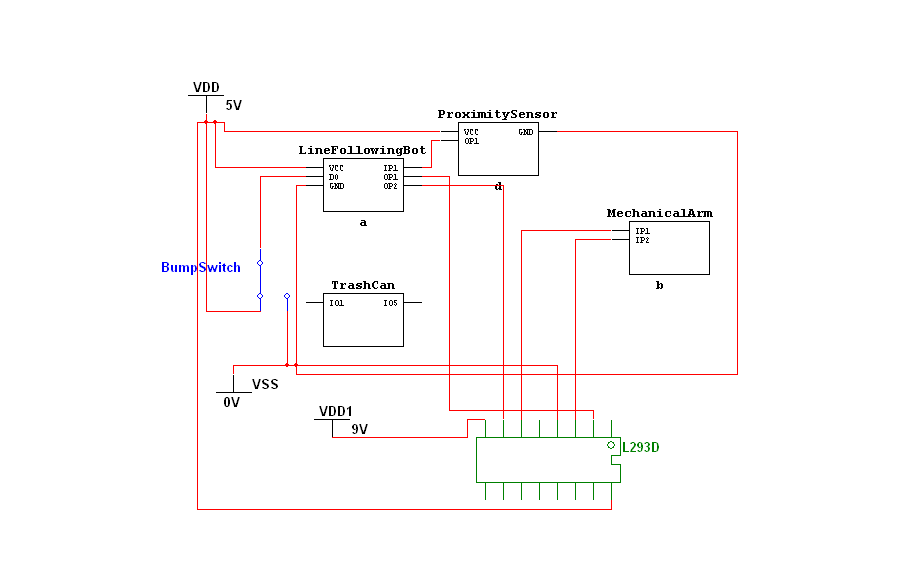
* Check if switch is pressed for 3 seconds
* if yes, send signal to atmega16
* Instruct motors to follow the line by taking inputs from IR sensors
* Detect main trash can
* Dispatch all the waste in main trash can
* Repeat the process until main trash can is full
* If main trash can is full send signal to municipal office on internet

**BLOCK DIAGRAM**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | The sensor | |  |  |  |
|  | gives logic | |  |  |  |
|  | high | |  |  |  |
| Trash Can |  | The can |  | Empties |  |
|  |  | itself in the |  |
| with sensor |  | travels |  |  |
|  |  | main trash |  |
| to detect if |  | automatically |  |  |
|  |  | can |  |
| it is full |  | to the main |  |  |
|  |  |  |  |
|  |  | can |  |  |  |
|  |  |  |  |  |



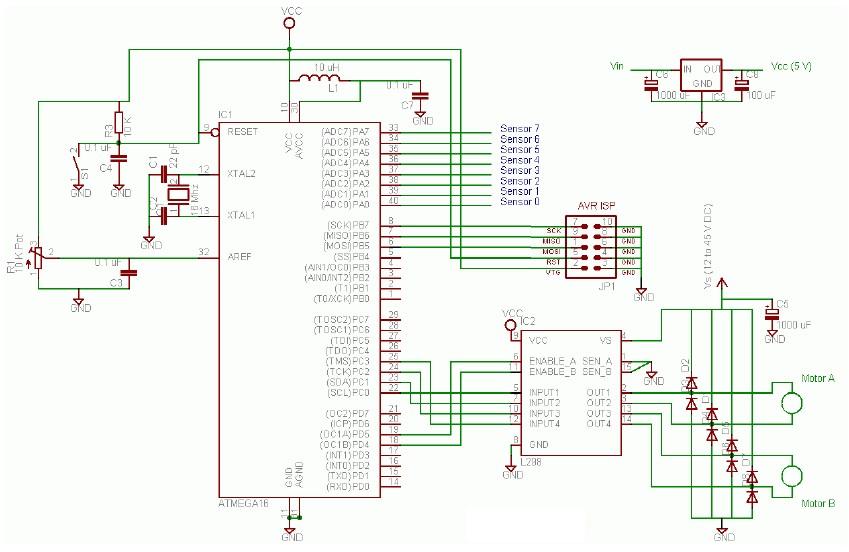
**1.3.2:CIRCUIT DIAGRAM**



**MECHANICAL ARM**



**LINE FOLLOWING BOT**



**1.4:COURIER AUTOMATED**

**1.4.1:Description**

We all have experienced the hardships of our all-important courier getting delivered at the time when we are not at home and the ensuing trips to the courier office. To address this problem we are implementing a smart courier receiving system integrated to the internet.

This module uses a 125KHz RFID reader (EM18), RaspberryPi (RPi), RFID tag, a motor to open the slit and an Android application **Instapush**. RFID tags will be provided to every authorized courier delivery person whose database will be stored in the RaspberryPi. When the delivery boy scans his RFID tag on the RFID reader, and on verification by the Raspberry PI, it will send a OneTimePassword (OTP) over the internet via the Instapush application to the delivery boy’s phone. He needs to be signed in to the Instapush account for receiving the OTP. Then he needs to type the OTP using a keypad which will be verified by the RPi. This extra level of security ensures that even if the RFID tag falls into wrong hands, the slit will not open unless the OTP is input correctly.

On verification of the OTP the RaspberryPi will open the slit and the courier boy can drop the box in the slit. A notification will also be send to the owner that a courier has been accepted. The slit will be open for only a short period of time after which the RPi will close the slit.

**Algorithm**:

* Create a list of authenticated values for RFID tags
* The courier boy shows RFID tag
* Take the value from RFID reader
* Search for the same value in the list

o If value found, send signal to slit in the door to open and receive the parcel

1. If the value doesn't matches with any value in the list display a message on LCD to come some other time

**BLOCK DIAGRAM**

Your Courier has been accepted

OTP entered by the courier boy is transmitted to RPi

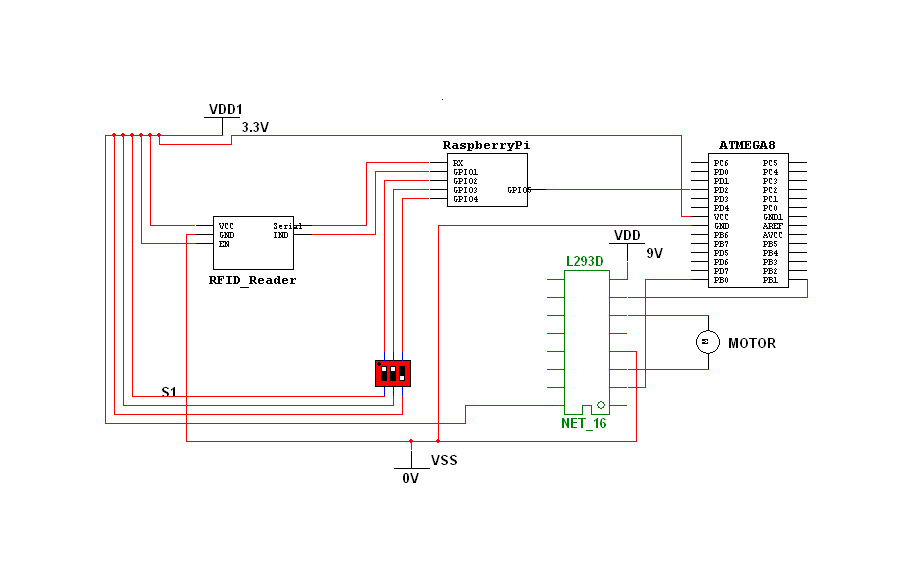
Response transmitted back to PI

On verification RPI sends a notification via internet

Raspberry PI

Courier Boy scans his ID

RPi sends an OTP over the internet

**1.4.2:CIRCUIT DIAGRAM**

**1.5:LEAKAGE ALERT**

**1.5.1:Description**

We read daily in the newspapers about fires taking place in houses due to Gas Leakage. All the valuable things in the house will get destroyed by a minor negligence. To avoid this, we are using a gas sensor integrated to the internet which will send a real time status to our phone.

In this module we are using a gas sensor (name), HT121E (encoder), HT121D(decoder), RF transmitter and receiver which are all powered by 3.3V source.

The gas sensor provides with an analog and digital output. The threshold of the gas sensor can be varied by using a potentiometer present on the sensor. Thus when the threshold is exceeded i.e gas leak occurs, the output goes low which is inverted using a not gate and send to HT12E which transmits the signal serially to the RF transmitter which is picked up by the RF receiver. The receiver in turn provides the signal to HT12D which is connected to the RPi and the latter sends a notification to the fire brigade about the gas leak so that preventive steps can be undertaken before a fire starts.

**Algorithm**:

* When there is a presence of LPG over the prescribed limit in the air gas sensor goes high indicating a gas leak
* It sends a signal to the Raspberry-Pi
  + The Raspberry-Pi sends a notification to our phone

**BLOCK DIAGRAM**

There is a gas leak in your apartment

Raspberry PI

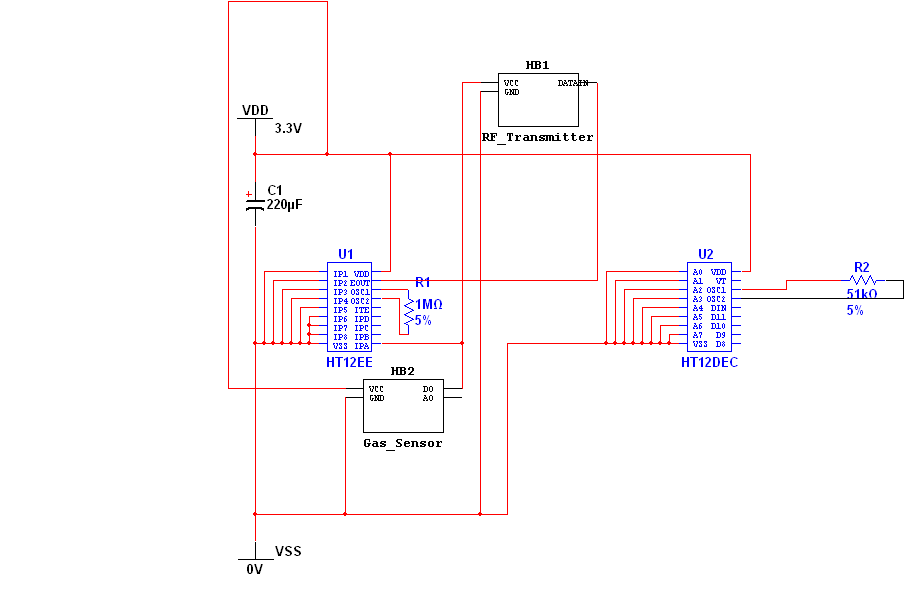
Radio Frequency Transmission

GAS

SENSOR

Sends notification over the internet to your phone

**1.5.2:CIRCUIT DIAGRAM**

****