CIA - II

Internet Of Things

Anuroopa. M
IV CSE A
MEUCS015

PART - B

10) **Arduino Board:**

★ **Microcontroller:**

The Atmega328p is the arduino brain. Everything on the arduino board is meant to support the microcontroller.

★ **Digital Pins:**

Arduino has 14 digital pins labelled from 0 to 13. that acts as both i/p and o/p. They can read two different states. high and LOW.

★ **PWM pins:**

These are digital pins marked with a (pins 11, 10, 9, 6, 5, 3). PWM stands for "Pulse with modulation".

★ **TX and RX pins:**

T stands for transmit and R stands for receive. These pins are used to communicate with computer.

* **Analog pins:**

Analog pins are labelled A0 to A5. They can read different amounts of voltage between 0 to 5V.

* **Power pins:**

Arduino has 3.3V or 5V supply. The pin labelled as GND are ground pins.

* **Reset button:**

When this button is pressed the program that is arrently being processed will start from begining.

* **USB Jack:**

Through a male USB A to USB B is how we upload programs from computer to arduino.

* **Power Jack:**

Power Jack is to connect a component to power up your arduino.

Arduino Interfaces:

Serial (UART) communication:

i) Every arduino board has atleast one UART.

ii) Serial communication on digital pins Rx and Tx via USB. Pin 0 and Pin 1 cannot be used for input and output

iii) Built in serial monitor can be used for communication by selecting same baud rate that is used in call to begin().

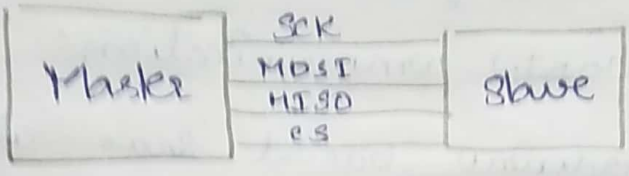
SPI communications:

i) SPI is a data protocol used by large microcontrollers for communication.

ii) With SPI there will be one master device that controls other peripheral devices.

iii) Devices has 3 lines in common. They are:

- MISO - Master In Slave Out. for sending data to master.
- MOSI - Master Out Slave In - sending data to peripherals.
- SCK - Serial clock.
- SS - Slave select pin on each device to enable and disable specific devices.



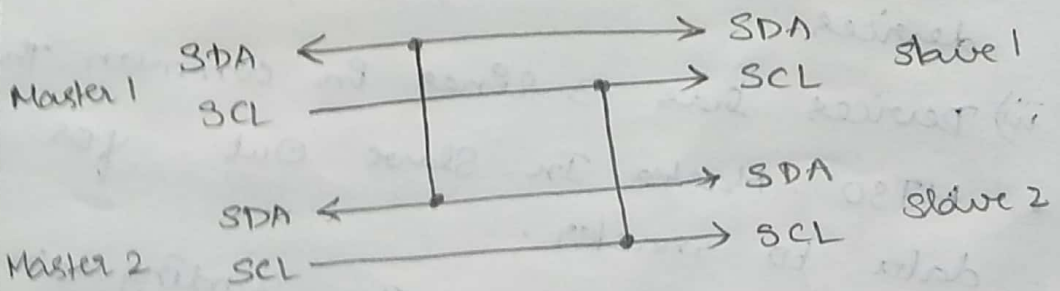
I^2C communications :

i) Inter-Integrated circuit or I^2C is one of the best protocol used when a workload of one Arduino is shared with another.

ii) The I2C protocol uses two lines to send and receive data.

iii) When the clock signal changes from LOW to HIGH, the address corresponds to specific device. and command is transferred from board to I2C device over SDA line.

iv) This information is sent bit by bit, then executes and transmits data back.



Characteristics	UART	I2C	SPI
Complexity	Simple	Easy to chain many devices	complex as device increases
Speed	slowest	Faster than UART	Fastest.
No. of devices	upto 2 devices.	upto 128 but may get complex as device incres	1 master & upto 5 slave devices
No. of wires	1	2	4
Duplex	Full Duplex	Half Duplex.	Full Duplex
No. of Masters & Slaves	No multiple slaves & masters.	Multiple slaves and masters.	Only one master but can have multiple slaves.

i) ii)

```
int sensorvalue = 0;
void setup () {
{
serial.begin (9600);
pinMode (13, OUTPUT);
}
void loop ()
{
sensorvalue = analogRead (A0);
digitalWrite (13, HIGH);
```

```
Serial.println(sensorvalue);
delay(sensorvalue); // wait for sensorvalue
                       milliseconds.

digitalWrite(13,LOW);
delay(sensorvalue); // wait for sensorvalue
                       milliseconds.
}
```

ii)

```
import smtplib;
import RPi.GPIO as GPIO.
import time.

# Email variables
SMTP_SERVER = 'smtp.gmail.com' # Email server
                                 (don't change).

SMTP_PORT = 587 # Server Port (don't change).
GMAIL_USERNAME = 'youremail@gmail.com'
GMAIL_PASSWORD = 'yourPassword'

# Set GPIO pins to use BCM pin numbers
GPIO.setmode(GPIO.BCM).
# Set digital pin 17 (BCM) to an input
GPIO.setup(17, GPIO.IN).
# Set digital pin 17 (BCM) to input and
enable pull up.
GPIO.setup(17, GPIO.IN, pull-up-down =
                        GPIO.PUD_UP).
```

```
# Event to detect button press
GPIO.add_event_detect(14, GPIO.FALLING)

class Emailer:
def sendmail(self, recipient, subject, content):

#Create Headers:
headers = ["From: " + GMAIL_USERNAME,
"Subject " + subject, "To " + recipient, "MIME-Version: 1.0"
"Content-Type: text/html"].
headers = "\r\n".join(headers)

#connect to gmail server:
session = smtplib.SMTP(SMTP_SERVER, SMTP_PORT)
session.ehlo()
session.starttls()
session.ehlo().

#Login to Gmail:
session.login(GMAIL_USERNAM, GMAIL_PASSWORD).

#send email & Exit.
session.sendmail(GMAIL_USERNAME, recipient
headers + "\r\n\r\n" + content).
session.quit.
sender = Emailer()

while True:
if GPIO.event_detected(14)
sendTo = 'anotheremail@email.com'
emailSubject = "Button Press Detected!"
emailContent = "The button has been pressed at:"
                time.ctime()
```

sender.sendemail (sentTo, emailSubject, emailContent)

print ("Email Sent");

time.sleep (0.1).

PART-A

1) Arduino sheilds are pre-built circuit boards used to connect to a number of arduino boards. These sheilds fit on top of arduino compatible boards to provide an additional capability like connecting to Internet, motor controlling providing wireless communicati LCD screen controlling etc. Different types of arduino sheilds are.

- ★ wireless sheild
- ★ GSM sheild.
- ★ Ethernet sheild.

2)

```
void setup ()
{
Serial.begin (9600);
pinMode (2, OUTPUT);
}
void loop ()
{
int value = Serial.read ();
if (value == '4')
{
```

```
{
 digitalWrite (2, HIGH);
}
else if (value == '6')
{
 digitalWrite (2, LOW);
}
}
```

3) The NodeMCU is an open source softwa and hardware development environment that is built around a very inexpen -sive system-on-a chip called ESP8266.

i) Arduino like device.
ii) Main component : ESP 8266.
iii) Programmable pins.
iv) Built-in wifi.
v) Power via USB.
vi) Low cost.

4) Smart Mobiles, smart refrigerators, smart fire alarm, smart watch, smart door lock, smart bicycle, medical sensor, fitness tracker, security system etc are examples of IOT devices.

5) delay: Delays are used to synchronize events.

setup: It is called only when arduino is powered on or reset. It is used to initialize variable and pin modes.

loop: The loop function runs continuously till the device is powered off. Logic is similar to while(1) for microcontroller programming.

```
void setup ()
{
  // setup code goes here, run once:
}
```

6) PROS:

→ Raspberry pi easily connects to Internet.
→ Pi has entire Linux software stack available.
→ can be programmed using variety of languages.

CONS:

→ Hardware access is not real time.
→ Lacks enough power to drive inductive loads.
→ Pi lacks an inbuilt analog to digital converter.
→ Hardware is not open source.

7) Insert the micro SD card into micro SD card reader. Then insert USB reader into Mac or PC. Rename card to 'Raspberry'. Make sure SD card is empty. When ready Raspberry ian OS will be written to SD card. Give Raspberry pi power by plugging power cord. It will start booting.
username: pi
password: raspberry

8) Baud rate : 1500 bauds.
Bit rate : 1500 * 4 = 6000 bps.

9) GPIO stands for General Purpose Input Output. Any of GPIO pins can be designated as an input or output pin and used for wide range of purposes. It is a way Raspberry pi can control and monitor outside world by being connected to electronic circuits. The raspberry pi is able to control LED's turning them on or off, or motors, or many other things with help of GPIO pins.