ROBOTICS CLUB
TRIBHUVAN UNIVERSITY
PULCHOWK CAMPUS

LOCUS 2021
18th National Technological Festival

Hardware Fellowship

DAY 5 AND 6

Overview

- Data Communication
- Communication Protocols
  - USART
  - I2C
  - SPI
- Interfacing Bluetooth Module with Arduino
- Interfacing MPU6050 with Arduino

Data Communication

- Communication between two devices
- Transfer data from one device to another

Things to consider....

- Representation of Data in device's memory
  - Int is stored 2 or 4 bytes
  - Strings are sequency of bytes
- Error Detection and Recovery
  - Parity Bits, CRC etc.
- Flow Control
  - Acknowledgements
- Line Coding
  - Convert 1's and 0's to appropriate voltage levels

Some More Things to Consider

- Serial vs Parallel Communication
- Bus vs. Point to Point
- Speed of Data Transfer
- Synchronous vs. Asynchronous

COMMUNICATION PROTOCOLS

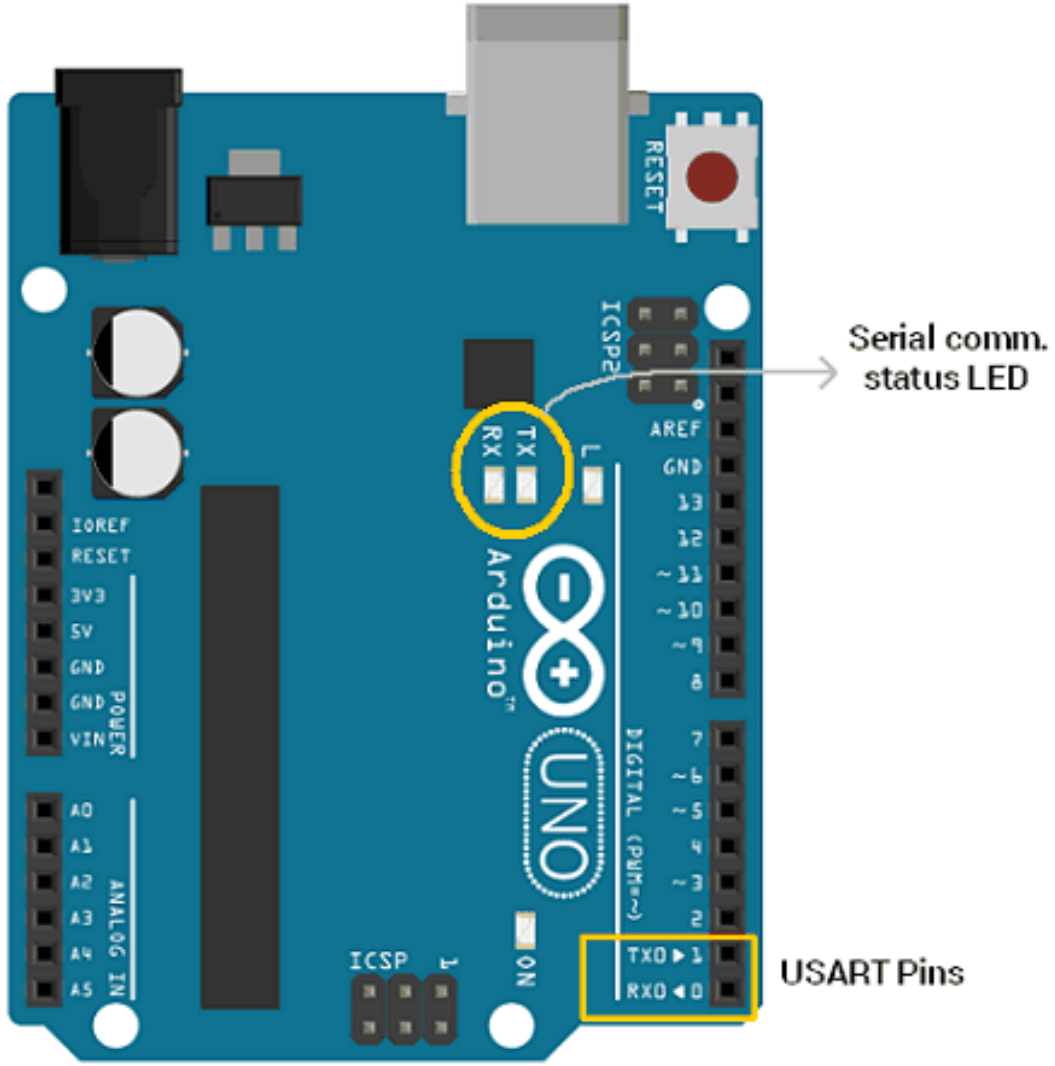
USART:

- Universal Synchronous Asynchronous Receiver and Transmitter
- It is a microchip that facilitates communication through a microprocessor's serial port using the RS-232C protocol.
- Takes data serially from peripheral -> converts into parallel data -> transmits it to the microcontroller and vice-versa
- Bit rate: rate of data transfer in serial data communication
- Baud rate: number of changes in signal per second.

•Serial communication over RX/TX pin uses TTL logic levels.

• Serial communication occurs between Arduino and computer/serial devices at a baud rate like 4800, 9600, 115200, 34800, etc.

•Arduino communicates with serial devices over digital pins 0 (RXD) and 1 (TXD). Whereas it communicates with PC/laptop over USB.



Arduino USART pin

- Serial Monitor: Arduino IDE has integrated serial monitor which can be used to watch serial data. After uploading program to the Arduino, open Serial Monitor to watch Serial data.

//Functions

- Serial.begin(baud rate): initiates the serial communication with given baud rate
- Serial.print(data, format type //optional): sends ASCII character on serial port so human can read it easily. This function converts the data to ascii character and then send (print) it.
- Serial.println(data, format type //optional):sends ASCII character on serial port followed by carriage return
- Serial.write():writes binary data to serial port
- Serial.available():returns the number of bytes available to read
- Serial.read():reads data serially
- Serial.readString():reads the received string

Sketch for reading serial string

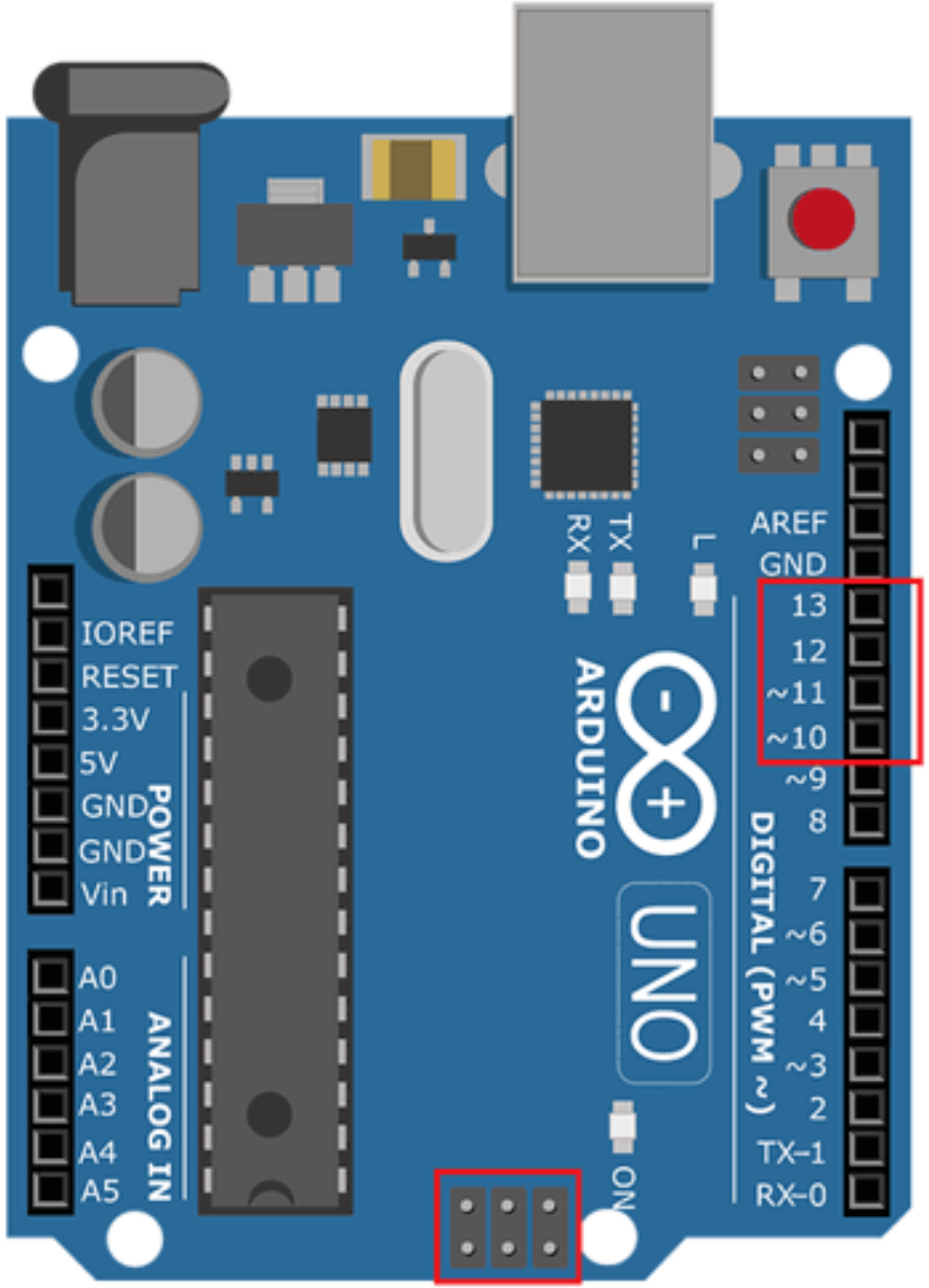
```
String receive_buffer;
void setup() {
  Serial.begin(9600);
}

void loop() {
  if(Serial.available()>0)  //check for any data received
  {
      receive_buffer = Serial.readString();  //read received data
      Serial.print("received data is: ");
      Serial.println(receive_buffer);        //display received data
  }
}
```

SPI:

- Serial Peripheral Interface
- Synchronous serial data protocol used by microcontrollers for communicating with its peripherals.
- With SPI connection, there is always a master which controls the peripheral devices
- SPI can be 3-line or 4-line interface.(MISO, MOSI, SCK, CS')
- MISO(Master In Slave Out): Slave line for sending data to master
- MOSI(Master Out Slave In): Master line for sending data to slave
- SCK(Serial Clock):clock pulses which synchronize data transmission generated by the master
- CS'(Chip Select):the pin on each device that the master can use to enable and disable specific devices

SPI Pins in Arduino Uno

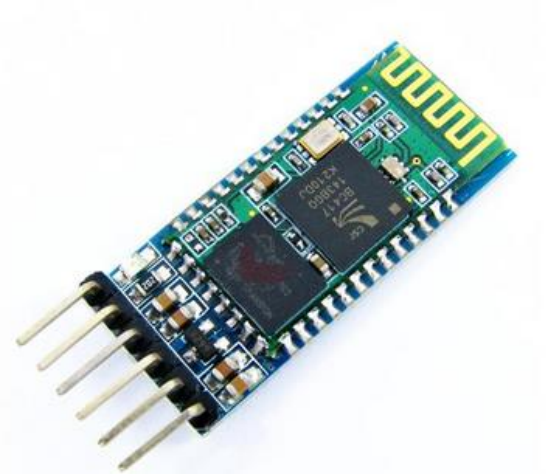
SPI Line	Pin in Arduino
MOSI	11 or ICSP-4
MISO	12 or ICSP-1
SCK	13 or ICSP-3
SS	10



BLUETOOTH MODULE HC-05

Introduction

- It is used for many applications like wireless headset, game controllers, wireless mouse, wireless keyboard and many more consumer applications.
- It has range up to <100m which depends upon transmitter and receiver, atmosphere, geographic & urban conditions.



Pin Description

It has 6 pins,

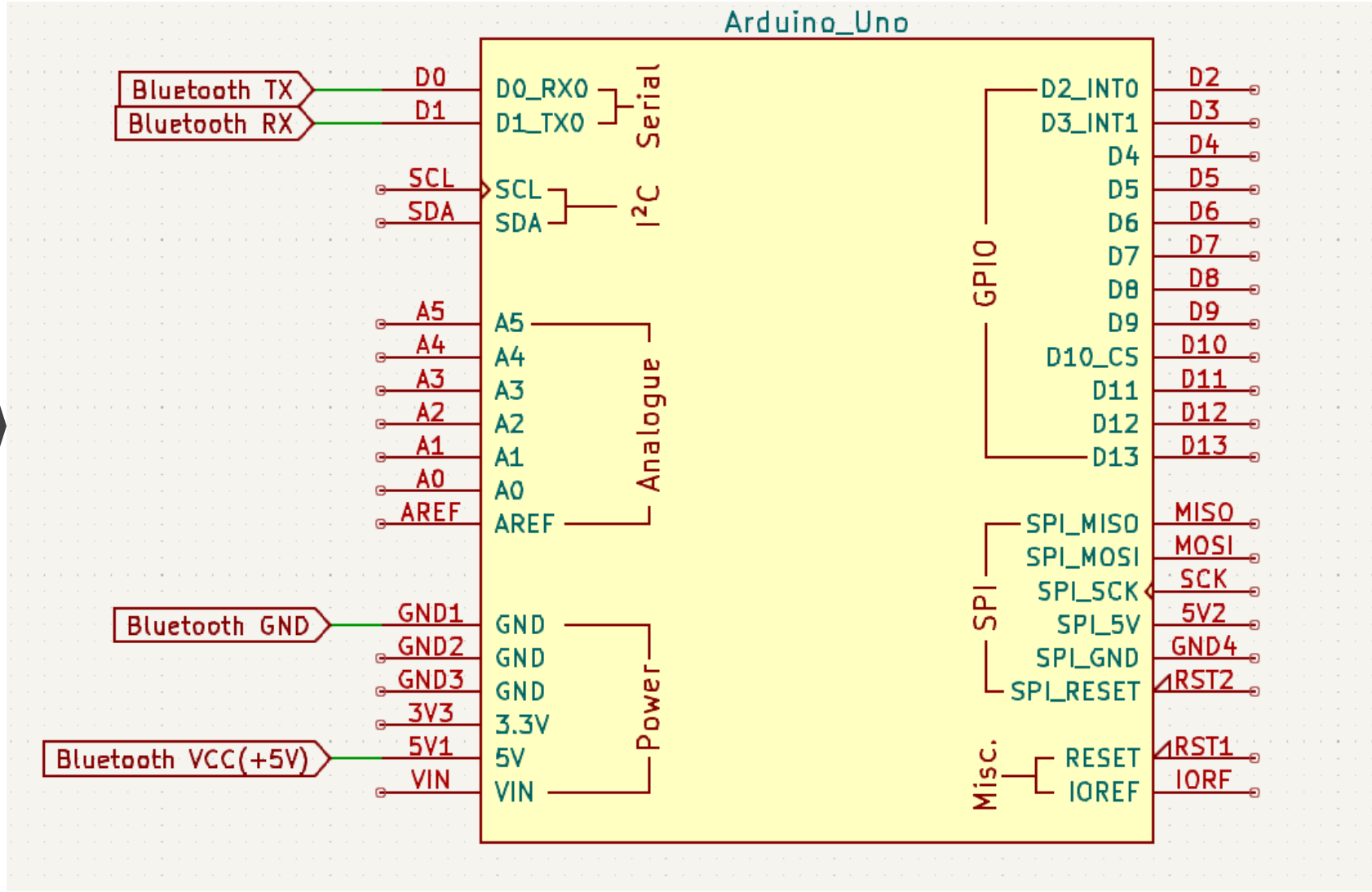
1. **Key/EN:** By default it is in data mode. The default baud rate of HC-05 in command mode is 38400bps and 9600 in data mode.

HC-05 module has two modes,

- **Data mode:** Exchange of data between devices.
- **Command mode:** It uses AT commands which are used to change setting of HC-05. To send these commands to module serial (USART) port is used.
- Set High to Trigger AT Command Mode
- Enable both NL and CR

- 2. **VCC:** Connect 5 V or 3.3 V to this Pin.
- 3. **GND:** Ground Pin of module.
- 4. **TXD:** Transmit Serial data (wirelessly received data by Bluetooth module transmitted out serially on TXD pin)
- 5. **RXD:** Receive data serially (received data will be transmitted wirelessly by Bluetooth module).
- 6. **State:** It tells whether module is connected or not

Interfacing Bluetooth Module (HC-05) with Arduino Uno



AT commands

- AT+NAME?
- AT+ROLE?
- AT+ADDR?
- AT+UART?

bluetooth

```
#include <SoftwareSerial.h>
#define L1 12
#define L2 10
#define R2 11
#define R1 9


char d = 'S';

SoftwareSerial mySerial (7, 8);  //(tx,rx

void setup() {
  // put your setup code here, to run once:
  pinMode(L1,OUTPUT);
  pinMode(R1,OUTPUT);
  pinMode(L2,OUTPUT);
  pinMode(R2,OUTPUT);

  Serial.begin(9600);
  mySerial.begin(9600);

}
```

```
void loop() {
  // put your main code here, to run repeatedly:

  if(mySerial.available()){
    d = mySerial.read();
    Serial.println(d);
  }

  switch(d){
    case 'F':
      digitalWrite(L1,HIGH);
      digitalWrite(R1,HIGH);
      digitalWrite(L2,LOW);
      digitalWrite(R2,LOW);
      break;

   case 'B':
      digitalWrite(L2,HIGH);
      digitalWrite(R2,HIGH);
      digitalWrite(L1,LOW);
      digitalWrite(R1,LOW);
      break;
```

```
    case 'R':
         digitalWrite(L1,HIGH);
         digitalWrite(R1,LOW);
         digitalWrite(L2,LOW);
         digitalWrite(R2,LOW);
         break;
     case 'L':
         digitalWrite(L1,LOW);
         digitalWrite(R1,HIGH);
         digitalWrite(L2,LOW);
         digitalWrite(R2,LOW);
         break;

     case 'S':
         digitalWrite(L1,LOW);
         digitalWrite(R1,LOW);
         digitalWrite(L2,LOW);
         digitalWrite(R2,LOW);
         break;

     default:
         break;
     }
}
```