Wire.h

Wire.h is for two wire interfaces, typically I2C. I2C uses the signals SDA(A4) & SCL(A5). I2C has 7 & 8 bit addresses, Wire.h uses 7 bit, 0-127.

```
void Wire.begin(address); // init wire library, join bus. If no address join as
master, else slave with address (7 bit)

Used by Master:

byte Wire.requestFrom(address, quantity[, stop]);
  // request data, 7 bit address
  // quantity bytes to request
  // stop = true sends stop, release bus (default)
  // stop = false restarts msg, keeps bus
  // bytes retrieved by available() and read()
  // returns no. bytes from slave

void Wire.beginTransmission(address);
  // transmit to slave device
  // queue bytes for sending by write() and transmit by endTransmission()

byte  Wire.endTransmission([stop]);
  // ends transmission started by begin...
  // transmits bytes queued by write()
  // stop = true sends stop, release bus (default)
  // stop = false restarts msg, keeps bus
  // byte 0 = success, 1 = data too long, 2 = rcvd NACK on address
  // 3 = recvd NACK on data, 4 = other error

byte Wire.write(value)
byte Wire.write(string);
byte Wire.write(data);
byte Wire.write(data, length);
  // value sends byte, string sends series of bytes, data is array of bytes
  // length is no. to send
  // returns no bytes written

byte Wire.available();
  // call on master after Wire.requestFrom(), or on slave inside Wire.onReceive()
  // returns no bytes available

byte Wire.read();
  // reads a byte, returns byte read (can cast to char, int etc)

void Wire.onReceive(function);
  // function to call when slave receives transmission
```

```
void Wire.onRequest(function);
  // function to call when master requests data from slave
```

Write

```
#include <Wire.h>

byte val = 0;

void setup()
{
  Wire.begin(); // join i2c bus on SDA(A4) & SCL(A5)
}

void loop()
{
  Wire.beginTransmission(44); // transmit to device #44 (0x2c)
                              // device address is specified in datasheet
  Wire.write(val); // sends byte
  Wire.endTransmission(); // stop transmitting

  val++;        // increment value
  if(val == 64) // if reached 64th position (max)
  {
    val = 0;    // start over from lowest value
  }
  delay(500);
}
```

Read

```
#include <Wire.h>

void setup()
{
  Wire.begin(); // join i2c bus (no address for master)
  Serial.begin(9600); // start serial for output
}

void loop()
{
  Wire.requestFrom(44, 6); // request 6 bytes from slave device #44

  while(Wire.available()) // while slave has more to send, up to 6
  {
    char c = Wire.read(); // receive a byte as character
    Serial.print(c); // print the character
  }

  delay(500);
}
```

Use of library

```
#include <Wire.h>

// 7 bit address 4 = 0x04 = B00000100 for device 1
#define DEV1 4

// 7 bit address 5 = 0x05 = B0000101 for device 2
#define DEV2 5

void setup()
{
  // ... code
  Wire.begin(); // init Wire library as i2C master (no address given)
  // ... code
}

// in general data transfers then proceed like this

Wire.beginTransmission(DEV1);
Wire.write(data);
  // could be 8 bit or sent in two bytes if 16 bit
  // can be register addresses then data to write to the device
  // ... more writes
Wire.endTransmission();

// to write data do this

Wire.beginTransmission(DEV1); // set up buffer
Wire.write(REGISTER or MEMORY ADDR);
Wire.endTransmission(); // transmit and end

Wire.beginTransmission(DEV1);
Wire.write(data);
Wire.endTransmission();

// to read data do this

Wire.beginTransmission(DEV1);
Wire.write(REGISTER or MEMORY ADDR);
Wire.endTransmission();

Wire.beginTransmission(DEV1);
Wire.requestFrom(DEV1, 1);
if(Wire.available)
{
  data = Wire.recevive();
byte = Wire.read(data);
Wire.endTransmission();
```