# Real-Time Embedded Systems

## Computer Assignment #1 Report

### Connection Protocols and Sensors


| | |
|---|---|
| Tina Behzad | 810195515 |
| Iman Mojtahed | 810195466 |
| Negar Kashef | 810195458 |


Winter 97

# Part 4:

## Questions

### 1.

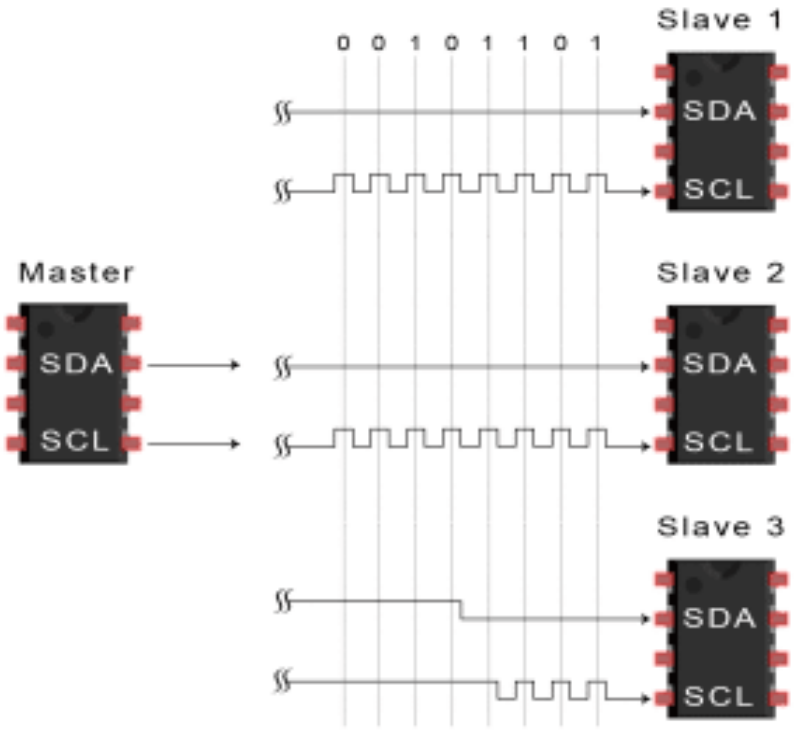
If you plan to use a multimaster device on a bus it is essential that all masters are multimasters. A single-master is simply a device, which does not understand the above mechanisms. If a singlemaster and a multimaster are connected, the singlemaster may well interrupt the multimaster causing unpredictable results.

If two devices start to communicate at the same time the one writing more zeros to the bus (or the slower device) wins the arbitration and the other device immediately discontinues any operation on the bus.

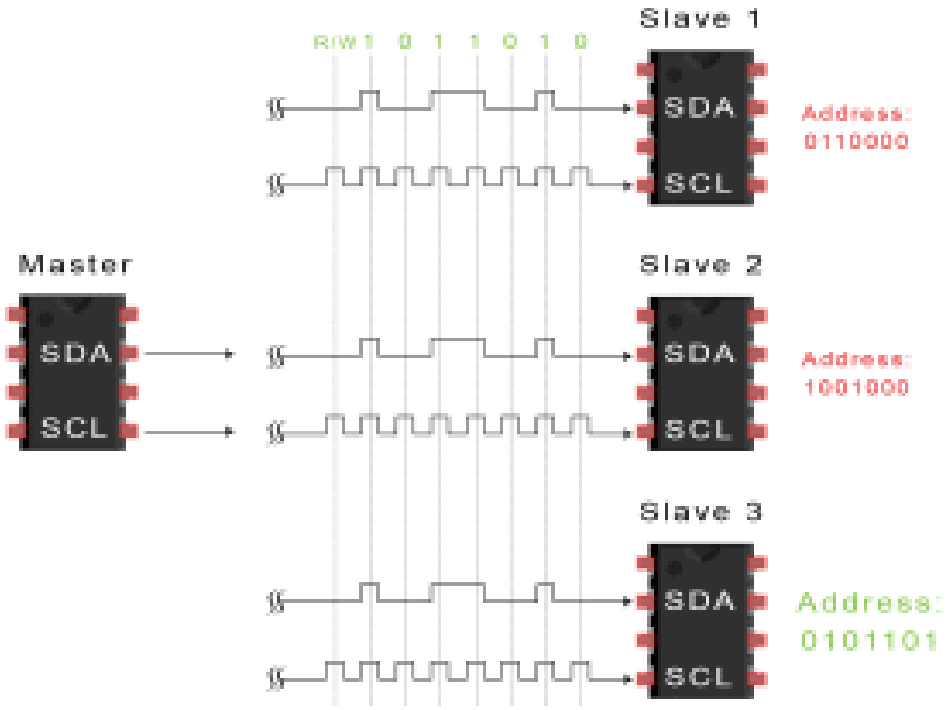
### 2.

Steps:

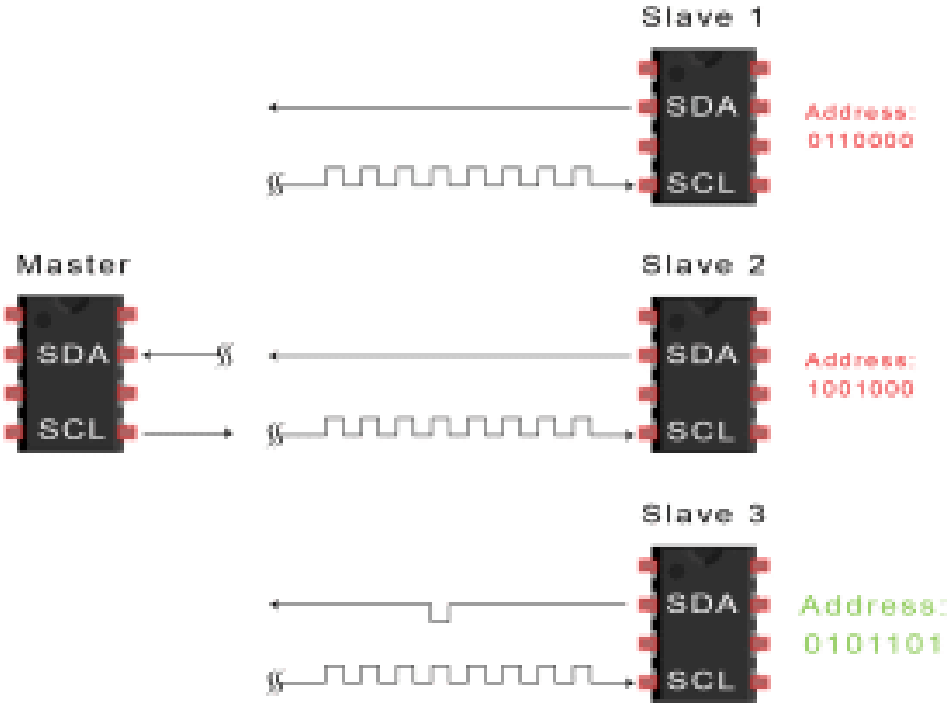
1. The master sends the start condition to every connected slave by switching the SDA line from a high voltage level to a low voltage level before switching the SCL line from high to low:



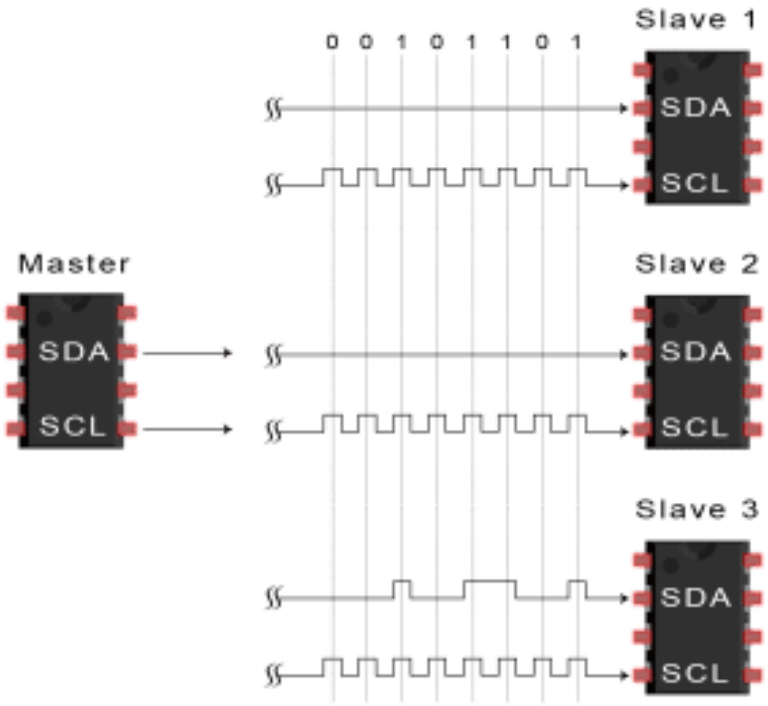
2. The master sends each slave the 7 or 10-bit address of the slave it wants to communicate with, along with the read/write bit:



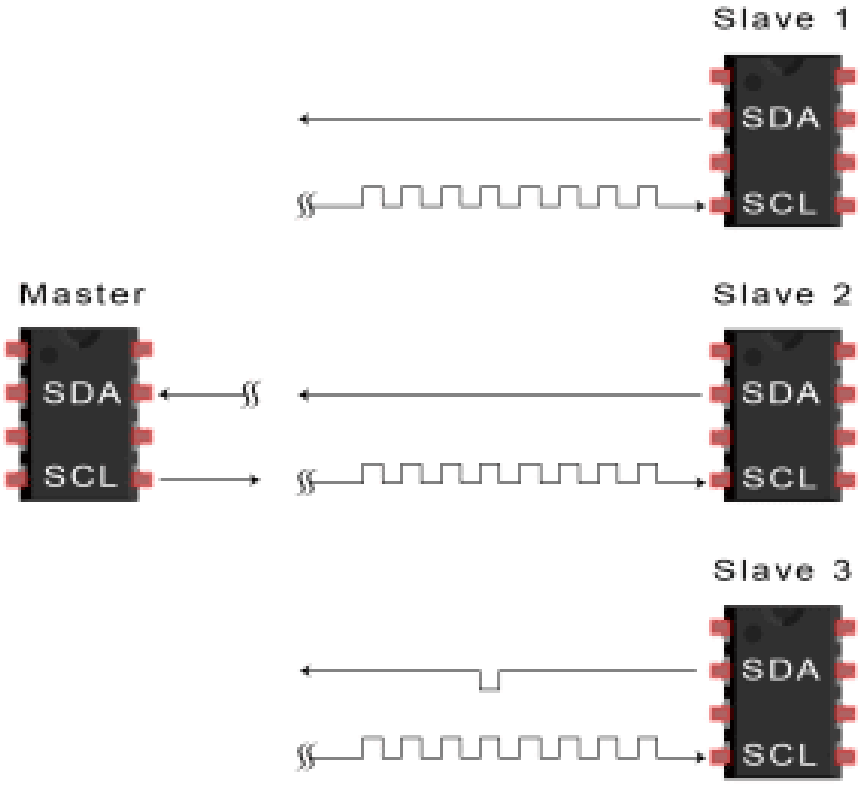

3. Each slave compares the address sent from the master to its own address. If the address matches, the slave returns an ACK bit by pulling the SDA line low for one bit. If the address from the master does not match the slave's own address, the slave leaves the SDA line high.



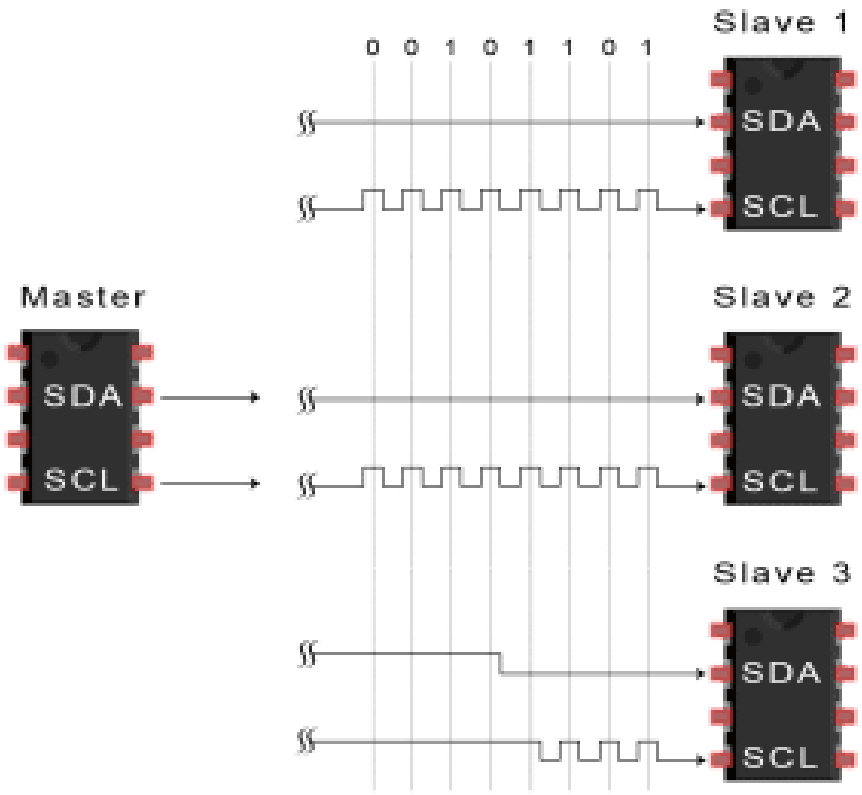
4. The master sends or receives the data frame:




5. After each data frame has been transferred, the receiving device returns another ACK bit to the sender to acknowledge successful receipt of the frame:



6. To stop the data transmission, the master sends a stop condition to the slave by switching SCL high before switching SDA high:




## 3.

Baud Rate

The baud rate specifies how fast data is sent over a serial line. It's usually expressed in units of bits-per-second (bps). If you invert the baud rate, you can find out just how long it takes to transmit a single bit. This value determines how long the transmitter holds a serial line high/low or at what period the receiving device samples its line.

Baud rates can be just about any value within reason. The only requirement is that both devices operate at the same rate. One of the more common baud rates, especially for simple stuff where speed isn't critical, is 9600 bps. Other "standard" baud are 1200, 2400, 4800, 19200, 38400, 57600, and 115200.

The higher a baud rate goes, the faster data is sent/received, but there are limits to how fast data can be transferred. You usually won't see speeds exceeding 115200 - that's fast for most microcontrollers. Get too high, and you'll begin to see errors on the receiving end, as clocks and sampling periods just can't keep up.

4.

| Features | UART | I2C |
|---|---|---|
| Pin Designations | TxD: Transmit Data<br>RxD: Receive Data | SDA: Serial Data<br>SCL: Serial Clock |
| Data rate | As this is is asynchronous communication, data rate between two devices wanting to communicate should be set to equal value. Maximum data rate supported is about 230 Kbps to 460kbps. | I2C supports 100 kbps, 400 kbps, 3.4 Mbps. Some variants also supports 10 Kbps and 1 Mbps. |
| Distance | Lower about 50 feet | Higher |
| Type of communication | Asynchronous | Synchronous |
| Number of masters | Not Application | One or more than One |
| Clock | No Common Clock signal is used. Both the devices will use there independent clocks. | There is common clock signal between multiple masters and multiple slaves. |
| Hardware complexity | lesser | more |

| | | |
|---|---|---|
| Software addressing | As this is one to one connection between two devices, addressing is not needed. | There will be multiple slaves and multiple masters and all masters can communicate with all the slaves. Upto 27 slave devices can be connected/addressed in the I2C interface circuit. |
| Advantages | • It is simple communication and most popular which is available due to UART support in almost all the devices with 9 pin connector. It is also referred as RS232 interface. | •Due to open collector design, limited slew rates can be achieved.<br>•More than one masters can be used in the electronic circuit design.<br>•Needs fewer i.e. only 2 wires for communication.<br>•I2C addressing is simple which does not require any CS lines used in SPI and it is easy to add extra devices on the bus.<br>•It uses open collector bus concept. Hence there is bus voltage flexibility on the interface bus.<br>•Uses flow control. |
| Disadvantages | • They are suitable for communication between only two devices.<br>• It supports fixed data rate agreed upon between devices initially before communication otherwise data will be garbled. | •Increases complexity of the circuit when number of slaves and masters increases.<br>•I2C interface is half duplex.<br>•Requires software stack to control the protocol and hence it needs some processing overheads on microcontroller/microprocessor. |

## 5.

Implementing serial communications involves hardware and software. The hardware provides the electrical signaling between Arduino and the device it is talking to. The software uses the hardware to send bytes or bits that the connected hardware understands.

The Arduino hardware has built-in support for serial communication on pins 0 and 1 (which also goes to the computer via the USB connection).

The SoftwareSerial library has been developed to allow serial communication on other digital pins of the Arduino, using software to replicate the functionality (hence the name "SoftwareSerial"). It is possible to have multiple software serial ports with speeds up to 115200 bps. A parameter enables inverted signaling for devices which require that protocol.

Limitations:

The library has the following known limitations:

- If using multiple software serial ports, only one can receive data at a time.
- Not all pins on the Mega and Mega 2560 support change interrupts, so only the following can be used for RX: 10, 11, 12, 13, 14, 15, 50, 51, 52, 53, A8 (62), A9 (63), A10 (64), A11 (65), A12 (66), A13 (67), A14 (68), A15 (69).
- Not all pins on the Leonardo and Micro support change interrupts, so only the following can be used for RX: 8, 9, 10, 11, 14 (MISO), 15 (SCK), 16 (MOSI).
- On Arduino or Genuino 101 the current maximum RX speed is 57600bps
- On Arduino or Genuino 101 RX doesn't work on Pin 13

## 6.

Ultrasonic distance sensor determines the distance to an object by measuring the time taken by the sound to reflect back from that object. The frequency of the sound is somewhere in the range of ultrasound; this ensures more concentrated direction of the sound wave because sound at higher frequency dissipates less in the environment. A typical ultrasonic distance sensor consists of two membranes. One membrane produces sound, another catches reflected echo. Basically they are speaker and microphone. The sound generator generates short (the length is a couple of periods) ultrasonic impulses and triggers the timer. Second membrane registers the arrival of the sound impulse and stops the timer. From the timer's time it is possible to calculate the distance traveled by the sound. The distance to the object is half of the distance traveled by the sound wave.

(It will reliably measure down to 3cm and will continue detecting down to 1cm or less but after 2-3cm the pulse width doesn't get any smaller Maximum range: little over 3m)

To summarize we can say:

- Range: tested from 5 to 200 cm

- Accuracy: absolute error ~0.035 cm/cm.
- Precision: standard deviation ~0.1-0.5 cm

## 7.

The waves may interfere with each other. Also one sensor may detect the sound sent by the other sensor not the reflect of its own sound.

## 8.

I2C is a bus, so like-named signals are connected together. The addressing scheme allows Arduino to select which device it's talking to so multiple sensors working with different addresses won't cause any problem.

Same sensors (when manufactured in the same company) usually have same addresses so connecting them via I2C will cause a problem, but in MPU 6050 The pin "AD0" selects between I2C address 0x68 and 0x69. That makes it possible to have two of these sensors in a project. Most breakout boards have a pull-up or pulldown resistor to make AD0 default low or high.

Connecting 3 or more of these sensors will normally be impossible but I2C bus can be extended with multiplexers in this case.

## 9.