

MIDDLESEX PLATFORM MEASURING SYSTEM

DESIGN DOCUMENT

MENG PROJECT 2018

SHIRWA HAJI M0044139

ANCILA ULUNGI M00517972

27TH MARCH 2018

**System Overview:**

The purpose of this project is used to check that the chair moves exactly in the specific direction according to commanded values therefore, it measures the dynamic orientation of chair at every 2s. We measure the translational motion of chair up to nearest 1 millimeter and angle up to ½ degree of accuracy.

The system involves the Middlesex 6DOF platform (The chair), motion sensors, an Arduino Mega board and an Uno board. The design consists of two major systems which are Monitor module and Sensor server which are interconnected to measure the accuracy of the chair.

In this design document the software and hardware designs will be discussed.

**Software design:**

the software consists of the two systems a Sensor server and Monitor module.

The monitor module is responsible for receiving data from the sensor server and comparing it with the commanded data and finally displaying the error.

Sensor Server:

The sensor server consists of two modules which are distance sensor which is used to measure the lengths of the actuator and the Angle server which is used to measure the pitch, roll and yaw values of the chair movement. The sensor server retrieves the distance and the angle values and sends it to the monitor to compare it with the commanded data and measure the error of the system.

Distance sensor:

The Distance sensor is a module that reads six distance sensor values using the Mega Arduino board. The mega reads the six distance sensor values using softwareSerial.

The format of sending the data is as it follows:

H xxxx, xxxx, xxxx, xxxx, xxxx, xxxx, \n

The format starts with an H, six length values separated by commas and ending with \n trailer and sends this to Uno on the specified port. XXXX represents integer values between 300mm and 1000mm. A value of zero will be sent if the sensor is not connected.

The first thing that the code does is to clear the strings and it adds a H as header. Then it reads the six sensor values and send it to the angle server. if a sensor is not connected then it will send a 0 value. The data will be sent as soon as it is available.

To test whether the distance sensors are connected a resistor is used to check if the pin is high (connected) or low (not connected) thus to stop the softwareserial ports from being on the floating status.

Angle server:

Angle server reads the 3 angular values using a jy901 library which provides a method which reads the sensor angle data. It calls the function “get angle” which is in the library and retrieves the roll pitch and yaw values.

The format of sending the combined data to the monitor is as follows:

H, xxxx, xxxx, xxxx, xxxx, xxxx, xxxx, yyyy, yyyy, yyyy, /n

Where the xxxx represents ASCI digits for the actuator lengths and the yyyy represents floating point numbers for the angles separating by commas and ending with \n trailer.

The first thing that the code does is to clear the strings and waits for character and if the character is a H then it read the distance data until it receive /n otherwise it waits until it receives H . once it receives /n it returns the sensor data. After that it combines with angular data and sends it to monitor. The system will send the combined data as soon as it is available.

## Harware design:

The hardware used for this system are mainly in the sensor server which consists of distance sensors, Arduino Mega and Arduino Uno and an angle monitor hardware module called jy901.

Distance Sensors:

The system is measuring the lengths of 6 actuator using Maxbotic sonar sensor MB1043 HRLV-MaxSonar EZ4. 1 sensor is attached to each actuator in such a way that the sensor is attached on one end of the actuator while a target is attached on the other end of the actuator. The sensors are capable of reading 30cm to 5000cm accurately precision up to the nearest 1mm on their serial pin.

To receive the values from the sensor, the sensor uses Pin 5-Serial Output. By default, the serial output is RS232 format (0 to Vcc) with a 1-mm resolution. If TTL output is desired, solder the TTL jumper pads on the back side of the PCB as shown in the photo to the right. For volume orders, the TTL option is available as no-cost factory installed jumper. The output is an ASCII capital “R”, followed by four ASCII character digits representing the range in millimeters, followed by a carriage return (ASCII 13). The maximum distance reported is 5000mm. The serial output is the most accurate of the range outputs. Serial data sent is 9600 baud, with 8 data bits, no parity, and one stop bit, as described from the datasheet of the sonar sensor

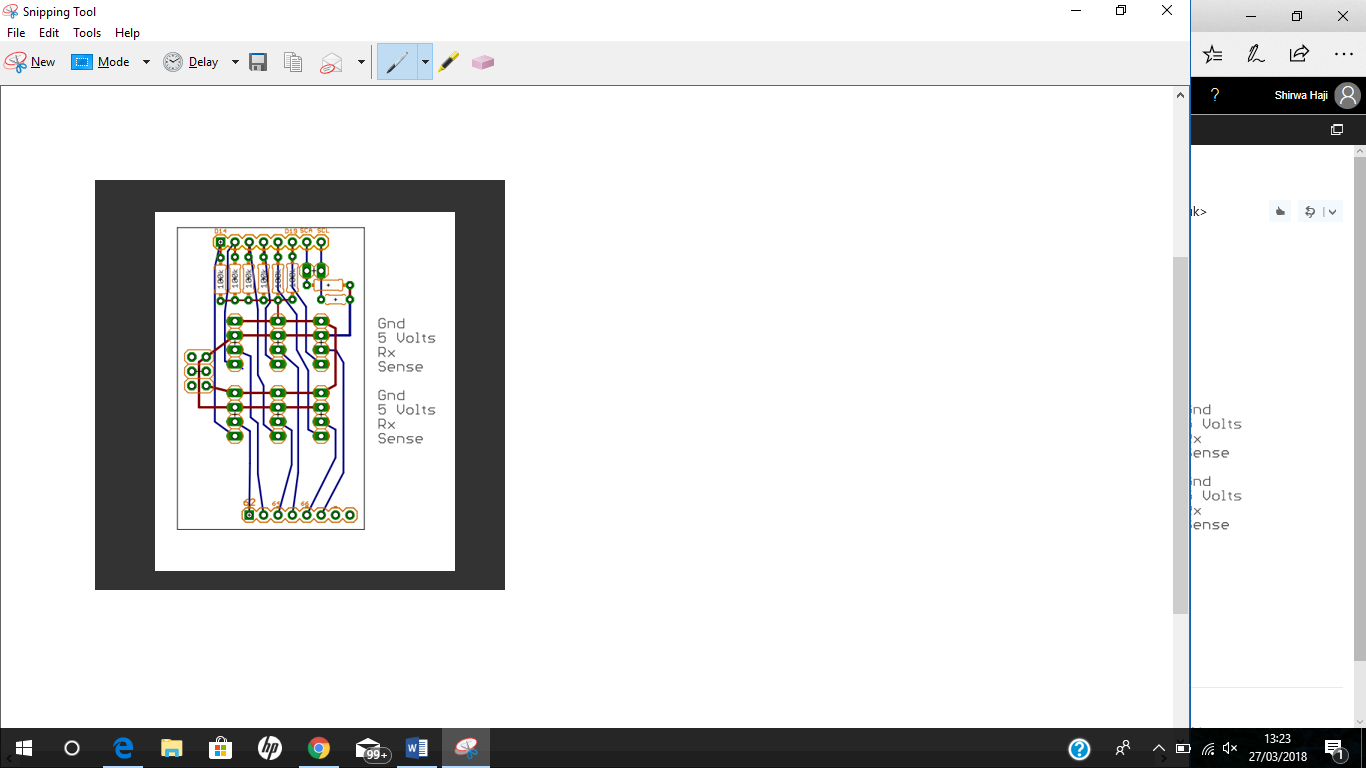
Multiple HRLV MaxSonar EZ sensors can be used simultaneously in the same environment generally with little to no interference (cross-talk). Even so, some cross-talk may still occur for users wishing to use many sensors in the same environment. This interference is rare and can be up to +/-1 cm of the target’s distance. Because of this, sensor to sensor interference must be accounted for. To avoid interference between sensors, chaining can be used to prevent cross-talk between sensors. This will be necessary when using 3+ sensors depending on mounting and environment. The chaining works in a way where it enables the first sensor and reads the data while disabling the rest of the sensors. Once it finishes reading the data the first sensor then gets disabled and the second sensor is enabled and reads the data while the other sensors are disabled and the chain goes on like this until all the sensors read the required data.

The recommended chaining method is an Output Commanded Loop. The first sensor will range, then trigger the next sensor to range and so on for all the sensors in the array. Once the last sensor has ranged, the array stops until the first sensor is triggered to range again.

Arduino Mega2560:

This is a microcontroller board based on ATmega2560 (data sheet). This board was selected because it has many number of pins and it can handle several sensors at the same time. For further details <https://store.arduino.cc/arduino-mega-2560-rev3>

To be able to connect the six sensors to the Mega board softwareserial ports were used because there weren’t enough hardware serial ports to connect all the six sensors to the Mega board. To connect all the six sensors to the softwareserial ports a distance sensor adapter board was created which makes it easier to connect the six sensors to the Mega board. This adapter board provides resistors used by the software to determine if the sensors are connected. Within the adapter there are six connectors one for each sensor that provides power, receive serial data and the fourth pin it tests whether the sensor is connected. Fig 1 shows the connections within the distance sensor adapter board.



**Figure 1**

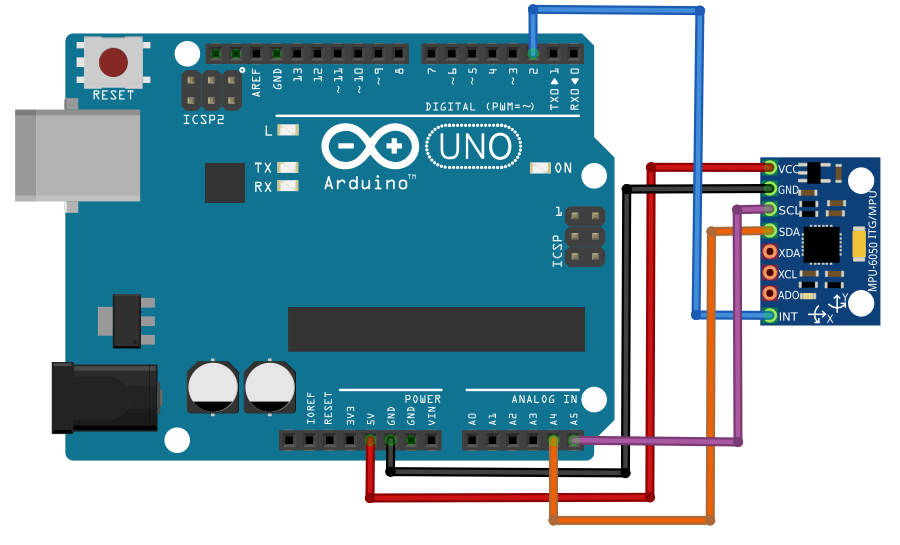
Arduino Uno:

This is a microcontroller board based on the ATmega328p (data sheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. For further detail <https://store.arduino.cc/usa/arduino-uno-rev3>

Within the Arduino Uno there is an angle monitor hardware module called jy901 which is used to measure the angles. It contains an integrated circuit called an mpu6050, this module reads angles in three degrees of freedom and rotations in three degrees of freedom. The information is provided using i2c module.

The jy901 module is connected to the Uno board through 4 connections of power, ground, i2c clock and i2c data.

Figure 2 shows the jy901 connected to the Arduino Uno board. In the figure it shows an interrupt connection in which it is not used in our design.



**Figure 2**

The connection between Mega and Uno is using softwareserial on port 10 for receiving the data using jumper wires.

Block diagram 1 shows how the hardware boards are connected to each other and what each board is involved with and finally sending the data to the monitor.

**Arduino Uno**

* Receives Six sensor values
* Get the angular measurements
* Arranging the data in seeding format
* Sending data to monitor

Monitor

**Arduino Mega**

* Receives the six sensor values
* Sending sensor values to R101

**Block Diagram 1**