**CSE 316**

**Microprocessor and Microcontroller Sessional**

**Project Documentation**

**Project: N-puzzle with Hand Gesture**

Preetom Saha Arko

Student No. 201305001

Chowdhury Rafeed Rahman

Student No. 201305025

**Block diagram:**

4-to-16 line decoder

IC-74154

4-to-16 line decoder

IC-74154

4-to-16 line decoder

IC-74154

4 8\*8 LED dot matrices

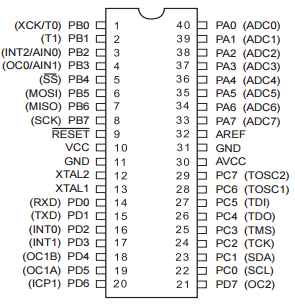
ATMega32

Gyroscope

MPU-6050 sensor

16\*2 character LCD module

**Components and Working principle:**

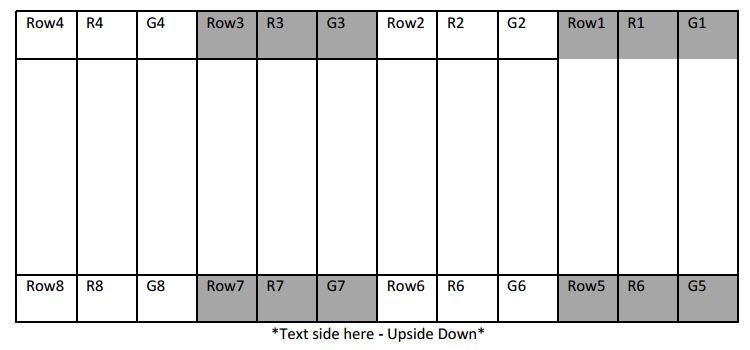
**ATMega32:**

ATMega32 has a total of 40 pins and 4 ports. Each port has 8 bits.

**Bi-color 8\*8 LED dot matrix:**

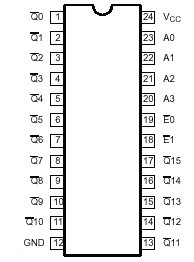
Each of the 64 positions (i.e., LED) can glow in green or red. There are 24 pins in total to select which LED(s) we want to light up. 8 of the pins are used to select the row(s), while the remaining 16 pins are used to select the column(s) (The notion of row and columns are only artificial and depends on the way we consider the orientation of the LED matrix. Rotating the matrix by 90 degrees will alter the  
orientation. However, we will use a fix orientation which is described later in this section). Among these 16 pins, 8 pins are used for lighting the LEDs red while the other 8 pins are used if we want to light the LEDs green. LED matrices can be oriented in two flavors: common (row) anode and common (row) cathode. The difference between these two configurations is how we light up a LED. With common anode orientation positive voltage is attached to rows and ground to columns. With common cathode the connections are reversed. In our project we will use the common anode orientation.

The pin diagram of the bicolor LED matrix is given below:



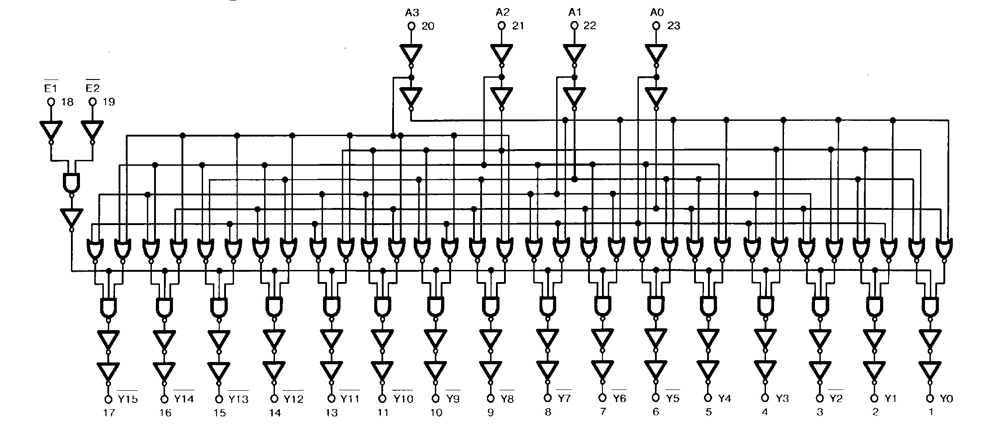
Description of pins:

1. Rows are anode (+ve is to be given). Rows are numbered from up to down.
2. Columns are cathode (GND is to be given). Columns are numbered from left to right.
3. RowX denotes row number.
4. GX denotes to light up green LED of column X.
5. RX denote to light up RED LED of column X.

**4-to-16 line decoder (IC-74154):**

4-to-16 line decoder has 4 inputs and 16 outputs. There are = 16 possible combinations of inputs, and for each input combination, only one output line will be low and the other output lines will be high (active low). Here A0, A1, A2 and A3 are the input lines and the ’s are the output lines. and are enable/strobe pins. A binary code applied to the four inputs (A0 to A3) provides a low level at the selected one of sixteen outputs excluding the other fifteen outputs, when both the strobe inputs, and , are held low. When either strobe input is held high, the decoding function is inhibited to keep all outputs high.

The logical circuit diagram of the decoder looks like this:



The decoder does the task of multiplexing for the 8\*8 LED dot matrices. We are using four 8\*8 LED dot matrices. So there will be 16 rows and 16 columns. A 4-to-16 line decoder will be used to multiplex the rows. We will multiplex the columns too. 16 pins for green columns will be multiplexed by a 4-to-16 line decoder and 16 pins for red columns will be multiplexed by another 4-to-16 line decoder.

set all rows to logic 0

for each row

set the row to logic 1

set all columns to logic 1

for each column

set the column to logic 0 if necessary

if the column was set to logic 0

set the column to logic 1

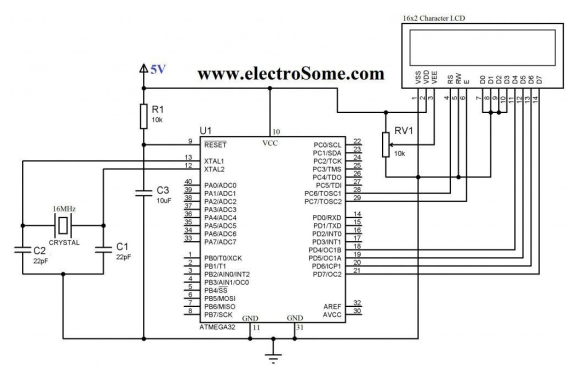
set the row to logic 0

If this is done fast enough, human eye will see the whole pattern together.

Without multiplexing the columns, we would require 32 + 4 = 36 pins at least, but there are only 32 pins in the 4 ports of ATMega32. Using decoder and multiplexing the columns, we can do this by using only 12 pins.

**16\*2 character LCD module:**

We will use the 4-bit mode. We will use the library file provided in this link: <https://electrosome.com/interfacing-lcd-atmega32-microcontroller-atmel-studio/> .



We do not need to use crystal. The connections of VSS, VDD, VEE, and D0-D3 are also not needed. In fact, the available LCD modules in the market have these connections built-in.

The connection summary is given in the following table.

|  |  |
| --- | --- |
| **LCD Module Pin** | **ATmega32 Pin** |
| RS | PC6 |
| RW | GND |
| E | PC7 |

|  |  |
| --- | --- |
| D4-D7 | D4-D7 |
| VCC | 5V |
| GND (both) | GND |

**Gyroscope (MPU 6050 sensor):**

SCL and SDA pins of gyroscope will be connected with the SCL and SDA pins of ATMega32 respectively. VCC and GND should be connected as usual. As we are not using multiple gyroscopes, the other pins will not be used.

We need to detect motions in x and y axis.

The MPU6050.h has two basic functions:

mpu6050\_init(); // start gyro

mpu6050\_getRawData(int16\_t \*ax, int16\_t \*ay, int16\_t \*az);

The last mentioned function populates the value in the registers ACCEL\_XOUT\_H, ACCEL\_XOUT\_L, ACCEL\_YOUT\_H, ACCEL\_YOUT\_L, ACCEL\_ZOUT\_H, ACCEL\_ZOUT\_L in 2’s complement form.