Project Documentation

Portable Gaming Console

Team Members :

1.Shruti Jain
2 . Siddharth Dangi
3.Chahat Bajpai

Team Mentors:

1. Sonu Agarwal
2. Shivendu Bhushan

Basic Aim:

Our basic objective is to fabricate a gaming console using arduino and TFT display.

Motivation:

The motivation for this project is drawn from the desire to work on hardware as well as on software. From the applications of Arduino we got this very fascinating idea of gaming consoles.
The coordinators finally agreed upon it and also encouraged it as this project would involve the use of Arduino Due which is very new to the world of Arduinos and had never been used in the club before.

Theory:

Our gaming console comprises of a controller and a display. To build up the controller we have primarily used Arduino Due, Bluetooth module, accelerometer and joystick while the display maing comprises of Bluetooth module, Arduino mega and TFT Shield. Let us look into the details of each:

Arduino Due :

Arduino Due is the latest known microcontroller board released on October 22, 2012. It is based on Atmel SAM3X8E ARM Cortex-M3 CPU . Its 32-bit ARM core microcontroller makes it first of its kind. It has 54 digital input/output pins (of which 12 can be used as PWM outputs), 12 analog inputs, 4 UARTs (hardware serial ports), a 84 MHz clock, an USB OTG capable connection, 2 DAC (digital to analog), 2 TWI, a power jack, an SPI header, a JTAG header, a reset button and an erase button.

Power:
Due can be powered by using either DC power jack(7-12V), the USB connector(5V) or the VIN pin of the board (7-12V). The board has 5V, 3.3V and GND pins generating respective outputs.

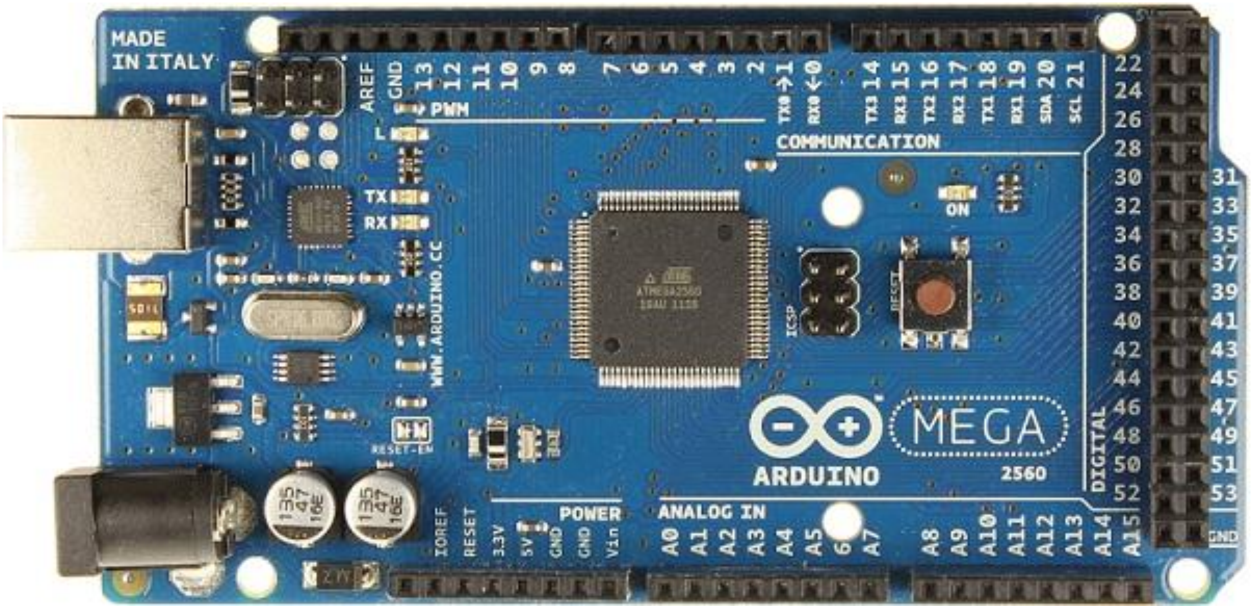
Memory:

The SAM3X has 512KB of flash memory for storing code(greatest of all) – again providing the board a distinguishing feature.

Input and Output

Due has 54 digital I/O pins operating at 3.3V. There are 4 set of Rx / Tx pins used for receiving and transferring data. The board also contains pins 2 to 13 which give PWM output, pins A0 to A11 which are for analog input, DAC1 /DAC2 which provide true analog output, reset pin etc.

<u>Arduino Mega:</u>



It is a microcontroller board based on Atmega1280. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs,
4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Atmega1280 has a flash memory of 128KB. Unlike Arduino Due, its operating voltage is 5V.

Bluetooth Module(RN-42):

This is a RN 42 FLY477 bluetooth module shield from Rhydolabz designed in accordance with arduino uno’s board design but can be mounted on other arduino boards also. The module is preset to work at a baud rate of 115200 , but can be set to any baud between 2400-115200 and has an operating voltage of 3.3V to 6V. The module settings can be changed by taking the module into command mode and using the commands given in the RN 42 manual.

Accelerometer:

The MMA 7361 is a 3- axis accelerometer. The IC require a very low power amount and is easy to use. It reads voltage along three axis relative to its initial position and maps it in Gs scale. Its features can be accessed using the library accelerometer.h

```
//                                        SL GS 0G   X  Y  Z
myAccelerometer.begin(3, 4, 5, A8, A10, A2);
//calibrating the accelerometer
myAccelerometer.calibrate();
//reads the values of your accelerometer
  myAccelerometer.read();
  Serial.println("The Accelerometer reads X");
  Serial.print(myAccelerometer._Xgs);
```

TFT Display:

Arduino 3.2'' TFT Touch Shield is an Arduino Mega compatible multicolored TFT display with a touch-screen and SD card socket. The TFT driver is based on SSD1289 with 16bit data and 4bit control interface. It can work with both 3.3V and 5V. It has a resolution of 240 x 320 pixels and supports UTFT library.

```
//include library
#include "UTFT.h"
//giving lcd specifications
UTFTtf myGLCD(ITDB32S, 38, 39, 40, 41);
myGLCD.InitLCD(LANDSCAPE);
```

Joystick :

Vcc
Xout
Yout
Sel
GND
SEL shorts to ground when pressed
X and Y act like 10K pots
Xout and Yout are Analog out

It is a 2 axis joystick i.e. it can be moved in 2-D plane. It has a potentiometer for each axis and each of them sends a voltage according to the movement along that axis which can be read at any of the analog input pins. This value ranges from 0 to 1023.

Speaker:

A small audio speaker that is ideal for radio and amplifier projects and is small enough to fit in robot projects.

Some Features:

- Small Size
- Power rating: 0.5W
- Impedance: 8 ohm

Overview

This project was initially named as Multiplayer Gaming Console but due to some events that followed its name ended up as Portable Gaming Console. We started our project with the plan of making two complete consoles (comprising of an arduino and a lcd each) and they were to interact via Bluetooth.

But few days after starting the project, we were told that we would be provided with only one lcd as it did not seem to be possible to get another lcd in such small time. Another major change took place when we realized that the lcd given to us is not compatible with Arduino Due and we will have to use Arduino Mega.

Our First Achievement:

The first task which we accomplished was the communication of Arduino with PC using Bluetooth. For this, since we were having the Bluetooth shield we mounted it over Arduino and used serial communication between them.

```
#include<SPI.h>
// start serial communication at 115200bps
 Serial.begin(115200);
```

Even after doing all connections properly, our Bluetooth module wasn't working. When we were scratching our heads at this, we realized that it was because the sliding switch on the Bluetooth was on usb side and not on Arduino side. This is a very frivolous mistake and should be taken care of.

Pre First Evaluation:

We spend hours trying to interface lcd with due but then finally we found that it isn't compatible with Due.

So finally we started using it with Arduino Mega and started coding our first game, which we called "The Brick Game". With the coding of this game we were exposed to the basic functions of the UTFT library.

```
#include<UTFT.h>
UTFT      myGLCD(ITDB32S, 38,39,40,41);      //gives specifications of lcd
myGLCD.InitLCD();     //intiates  lcd
myGLCD.clrScr();        //clears screen
myGLCD.setColor(VGA_RED);      //sets color to be red
myGLCD.fillScr(VGA_BLACK);      //fill screen with black color
 myGLCD.print("hello",27,212);       //prints hello
myGLCD.drawLine(x1,y1,x2,y2);      //draws line between (x1,y1) and (x2,y2)
myGLCD.drawRect(x1,y2,x2,y2); // for open rectangles
myGLCD.fillRect(x1,y2,x2,y2); // for filled rectangles
myGLCD.drawRoundRect(x1,y2,x2,y2); // for open rectangles
myGLCD.fillRoundRect(x1,y2,x2,y2); // for filled rectangles
```

We accessed the touch feature of LCD using the UTouch library. Some of its basic commands are:

```
#include <UTouch.h>
UTouch     myTouch(6,5,4,3,2);
myTouch.InitTouch();             //Initialize the touch screen
myTouch.dataAvailable()          //checks if data is available
myTouch.read();
     int x=myTouch.getX();
     int y=myTouch.getY();       //reading the coordinates of touch
```

So by the time of the first evaluation we had our first game working and serial communication done between PC and Arduino using bluetooth.

<u>Post First Evaluation :</u>

Since our first evaluation passed as per the expectations, our spirits were higher and we started work with even more vigor. But now a major hurdle was awaiting us. We were being told that we will have to share the display with one of our contemporaries until a new lcd comes(which actually never came – we shared that lcd even for the final presentation). This had serious repercussions as our working hours reduced significantly as at all times, atleast one of us was working with the code and needed lcd to check it.

SD CARD:

Soon after the first evaluation our first achievement was loading data from sd card. Initially we wanted to use it for loading the entire gaming code for which we were searching for a Bootloader that could load the code from SD card but after a lot of googling and posting on forums we got this response from a very experienced fellow:

-"I have a project in which I have to load sketch from the SD card instead of uploading from the Arduino IDE. Means the user should be able to choose from different pre compiled and verified codes and the code he selects should get uploaded... Can anyone help me in this context... Plzzz "

-"AFAIK that cannot be done."

So finally after this we decided that we will be storing the code in the flash memory of Mega and as for images we 'll be streaming them from the SD card where they 'll be stored in raw format. Also for the games we are storing the high scores in the SD card.

Drawing Bitmap:

Images can be easily drawn on the lcd using the bitmap arrays. The function in UTFT library is:

```
myGLCD.drawBitmap (x, y, height, width, name_array);
```

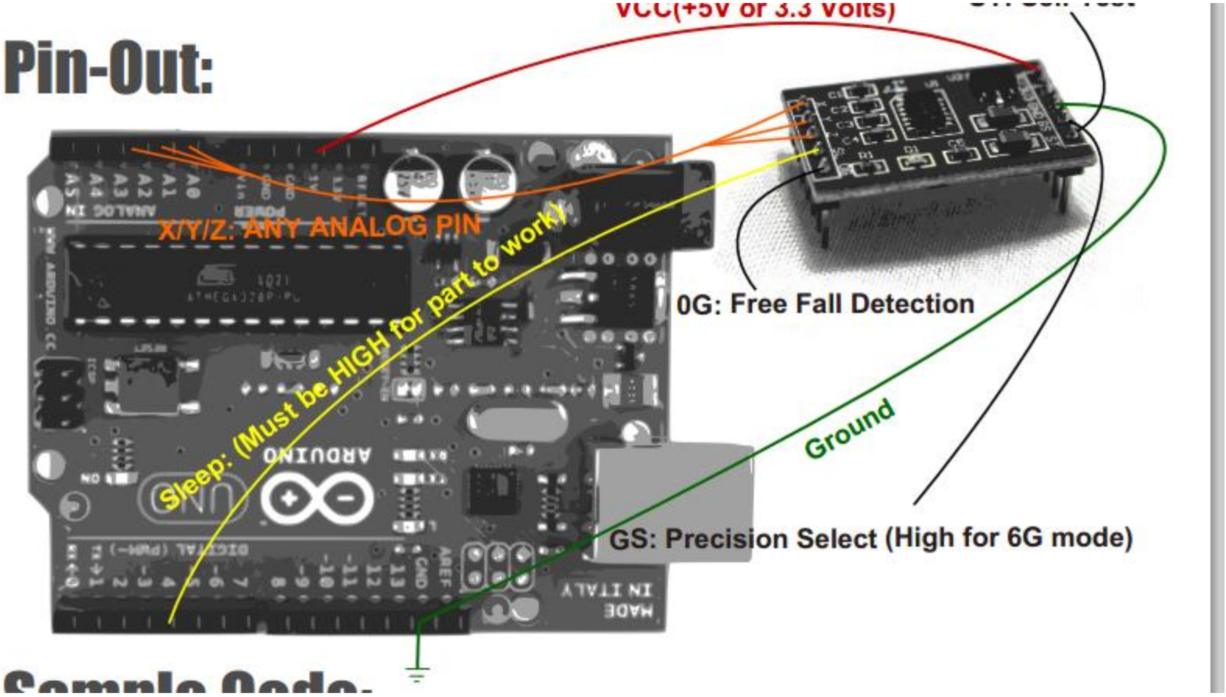
The array is nothing but the c file of the .png , .gif or .jpg files. There is an online converter available for this which can be accessed from *http://www.henningkarlsen.com/electronics/t_imageconverter565.php*.

Second Game:

As for second game initially we were thinking to implement Mortal Combat. But then we decided to make a kind of car racing game this could be controlled not just by touch but also by motion sensing and joystick.

Accelerometer:

The first game we implemented was by using touch. In order to play this game by motion sensing we used accelerometer. The accelerometer has a rather easy pin-out. We need to make only 5 connections – GND,VCC and three analog input pins for x, y and z axis respectively.



These analog input pins receive data in the terms of Gs. We are using it only for one-axis motion sensing.

<u>Joystick:</u>

Its usage was also quite easy. We used it for motion along 2-D.

In addition to the features mentioned above we have also added “Select” and ”Pause” buttons to the controller.

What have we come up with?

<u>Controller:</u>

It has following features:

1. It is portable and light.
2. It can be used to play games from distance upto 30 feet.
3. It can be used to control games using a joystick.
4. It also possess select and pause switches.

5. It can also be used to play games through motion sensing.
6. Switching between motion sensing and joystick is convenient with the sliding switch we have used with which accelerometer can be turned off. And hence we can play the game in any lying position also.
7. There is a power switch also which makes it easy to turn on/off the controller.

Game Pad:
It consists of display mounted over Arduino Mega for displaying the game.

A Word of Thanks

We would like to thank our mentors Sonu Agarwal and Shivendu Bhushan along with Swapnil Upadhyay for their incessant support, patience, suggestions and their believe in us .They inspired us to learn and explore a lot from this exciting project and checked the progress its regularly. They showed us the way at the times we were frustrated. Also our special thanks to Lohit Jain for his support on Bluetooth.

Moreover, it was fun doing this project in first years summers as we got to learn so much, we actually learned a new programming language and a set of new tools in it for creating exciting games.