

# USB Keyboard interfacing with arduino mega adk

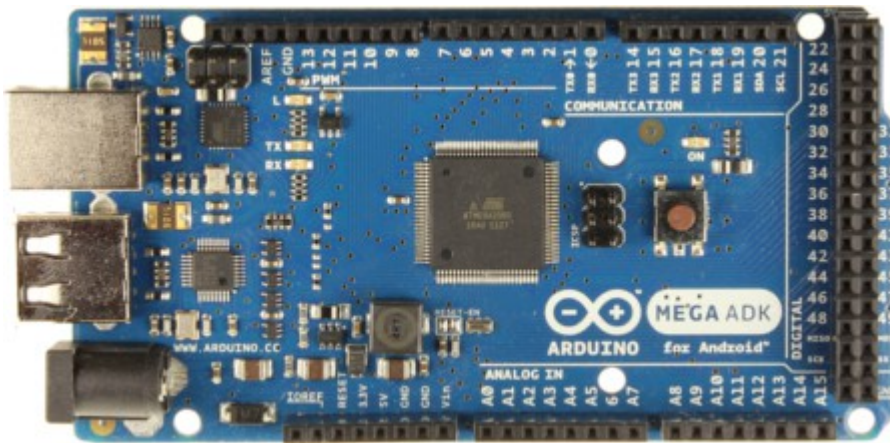
By :-Aashish Gupta

this tutorial describes the way of interfacing usb keyboard with arduino mega adk and displaying the output on laptop screen .

first let me give some information about the hardware and from where you can purchase it.

Following is the image of mega adk board for android. You can purchase this board by ordering it online on **www.arduino.cc** website or you can also purchase it from various distributors of arduino boards

you can get the list of distributors from above mentioned website only



The Arduino ADK is a microcontroller board based on the ATmega2560 . It has a USB host interface to connect with Android based phones, based on the MAX3421e IC. 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. Following is the link for its datasheet , if someone want to refer it in case

<http://www.atmel.com/Images/doc2549.pdf>

apart from above , one of the main speciality which is of all arduino boards is that they got their own software for coding and programming it . This software supports a vast library and variety of inbuilt functions which makes its coding quite comfortable as compared to normal c coding in code vision avr for atmega 8/16 chip.

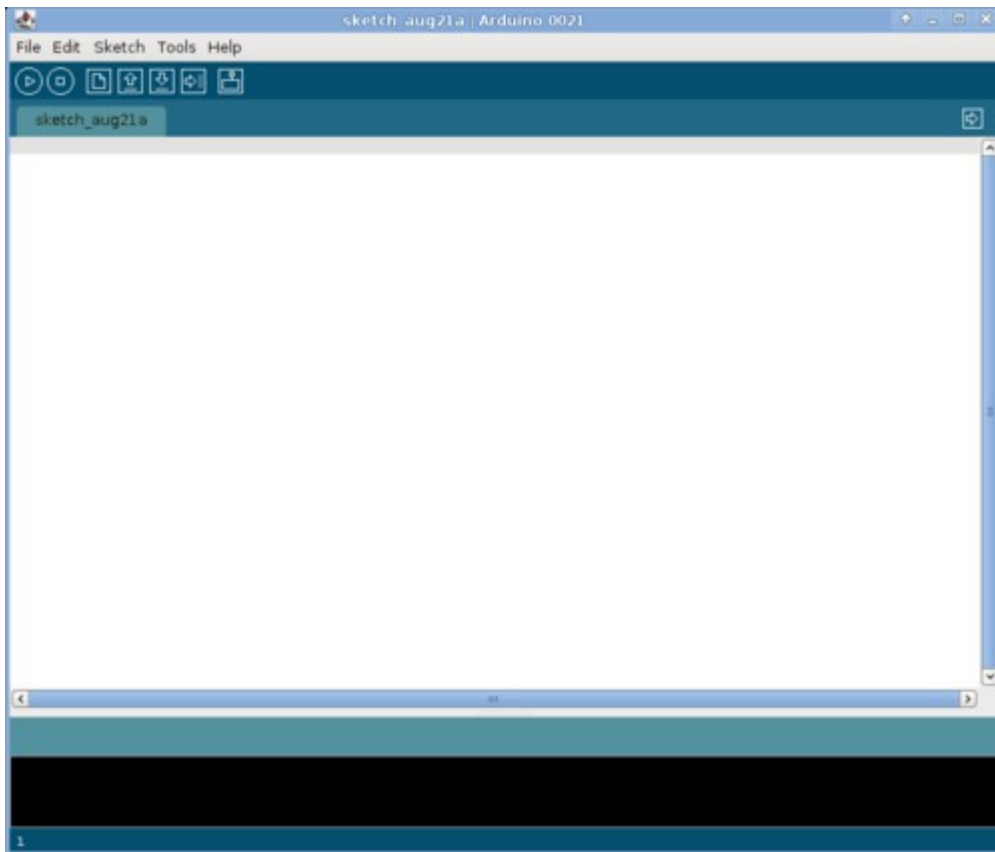
This software can be downloaded from arduino.cc website only . This software is available for windows as well as linux and mac. After installing it ,we can code in that software and compile It .

In order to upload your code connect your arduino with laptop's usb port using usb cable which comes

along with it , it will automatically install the drivers as soon as you connect , if it doesnt do so , then you can manually install them by browsing in the window which appear asking for driver. Drivers are available in arduino folder only inside c drive or whichever drive in which you have downloaded it. Just browse through those drivers and arduino drivers are installed. After that select your type of board and com port from tools . For more information over its installation you can refer to following youtube link

<http://www.youtube.com/watch?v=9PUBfliMZZk>

## Arduino software screen looks like as shown below



2 buttons which are below file tab are used to compile and upload code.

You can try some basic example code for e.g. select file → examples → basics → blink

compile and upload it, led on your arduino board will start blinking

## some basic features of arduino coding

code mainly consist of header files , self created functions and their definition and two special function void setup and void loop

1. void setup is that function which is executed in very beginning of program and is basically used to initialize variables and check basic initial conditions . Void loop is that function which is executed after void setup and it keeps on executing again and again, just like while(1) loop in normal c coding in code visionavr

2. serial.print() command is used for directly putting the variable inside the parenthesis on serial monitor screen

## **Now we come to actual interfacing of keyboard .**

Hardware part simply involves joining usb keyboard with usb shield and connecting arduino with laptop  
but for coding part we need to understand usb data packet form which I am giving below

USB data is sent in packets Least Significant Bit (LSB) first.

There are 4 main USB packet types :Token, Data, Handshake and Start of Frame.

Each packet is constructed from different field types, namely SYNC, PID, Address, Data, Endpoint, CRC and EOP.

The packets are then bundled into frames to create a USB message

The USB token packet is used to access the correct address and endpoint. It is constructed with the SYNC, PID, an 8 bit PID field, followed by a 7 bit address, followed by a 4 bit endpoint and a 5 bit CRC.

Both the address and endpoint field must be correctly decoded for correct operation.

The data packet may be of variable length, dependent upon the data. However, the data field will be an integral number of bytes.

### **Arduino code developed for interfacing is as follow**

```
#include <avr/pgmspace.h>
```

```
#include <avrpins.h>
```

```
#include <max3421e.h>
```

```
#include <usbhost.h>
```

```
#include <usb_ch9.h>
```

```
#include <Usb.h>
```

```
#include <usbhub.h>
```

```
#include <avr/pgmspace.h>
```

```
#include <address.h>
```

```
#include <hidboot.h>
```

```
#include <printhex.h>
```

```
#include <message.h>
```

```
#include <hexdump.h>
```

```
#include <parsetools.h>
```

```

class KbdRptParser : public KeyboardReportParser
{
    void PrintKey(uint8_t mod, uint8_t key);

protected:
    virtual void OnKeyDown    (uint8_t mod, uint8_t key);
    virtual void OnKeyUp      (uint8_t mod, uint8_t key);
    virtual void OnKeyPressed(uint8_t key);
};

void KbdRptParser::PrintKey(uint8_t m, uint8_t key)
{
    MODIFIERKEYS mod;
    *((uint8_t*)&mod) = m;
    Serial.print((mod.bmLeftCtrl  == 1) ? "C" : " ");
    Serial.print((mod.bmLeftShift == 1) ? "S" : " ");
    Serial.print((mod.bmLeftAlt   == 1) ? "A" : " ");
    Serial.print((mod.bmLeftGUI   == 1) ? "G" : " ");

    Serial.print(">");
    PrintHex<uint8_t>(key);
    Serial.print("< ");

    Serial.print((mod.bmRightCtrl  == 1) ? "C" : " ");
    Serial.print((mod.bmRightShift == 1) ? "S" : " ");
    Serial.print((mod.bmRightAlt   == 1) ? "A" : " ");
    Serial.println((mod.bmRightGUI  == 1) ? "G" : " ");
};

void KbdRptParser::OnKeyDown(uint8_t mod, uint8_t key)
{
    Serial.print("DN ");
    PrintKey(mod, key);
    uint8_t c = OemToAscii(mod, key);

    if (c)
        OnKeyPressed(c);
}

void KbdRptParser::OnKeyUp(uint8_t mod, uint8_t key)
{
    Serial.print("UP ");
    PrintKey(mod, key);
}

void KbdRptParser::OnKeyPressed(uint8_t key)
{
    Serial.print("ASCII: ");
    Serial.println((char)key);
}

```

```

};

USB  Usb;
//USBHub  Hub(&Usb);
HIDBoot<HID_PROTOCOL_KEYBOARD>  Keyboard(&Usb);

uint32_t next_time;

KbdRptParser Prs;

void setup()
{
  Serial.begin( 115200 );
  Serial.println("Start");

  if (Usb.Init() == -1)
    Serial.println("OSC did not start.");

  delay( 200 );

  next_time = millis() + 5000;

  Keyboard.SetReportParser(0, (HIDReportParser*)&Prs);
}

void loop()
{
  Usb.Task();
}

```

### **some imporant point for running code properly**

we need to download usb hostshield rev 2.0 version library from [www.github.com](http://www.github.com) for various header files included in code for using special functions. one more important thing which we need to do **after downloading the library is to uncomment a #define code written inside avrpin library's H file.**

**This uncommenting is required only if we are using Arduino Mega ADK board with MAX3421e built-in.**

### **some explanation of code**

usb is instance of USB class which need to be invoked every time as it calls other function onkeydown function is called when any key is pressed on keyboard, this function converts key (a particular code assigned to each key of keyboard while transferring its data using usb protocol) into ascii and also checks for special keys like alt, contol, shift by calling printkey function, it then calls onkeypressed function which puts the character in its original form onto the screen of serial monitor

using serial.print function .onkeyup function is called when key is released . Onkeyup and onkeydown together determines that a key has been pressed and onkeypress function prints it on serial monitor

## Getting output

After doing above mentioned things and successfully compiling the code and uploading it in arduino , we have to open serial monitor by clicking on the top right of arduino software window to see the output of any input from keyboard.

## Lcd interfacing with arduino

We can also connect lcd screen to arduino and obtain our display on lcd screen rather than on serial monitor.

For this we also need to make certain changes in code .

First we need to include liquid crystal library and take an instance of class liquid crystal giving no of pin on the board on which we have connected it .

We have to add lcd.begin() in void setup loop and

wherever we have to print anything we have to use lcd.print() command (where I have taken lcd as name of instance, you may take any other name). Following image shows properly working lcd screen interfaced with keyboard via arduino and keyboard is printing directly on lcd screen!!

