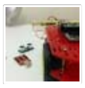
Popular Recent Comments Tags

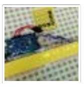
A Keypad Activated Interrupt
September 5, 2014

LEDs Sweeping Effect
July 17, 2014

Controlling a 2-Wheels Car
September 27, 2014



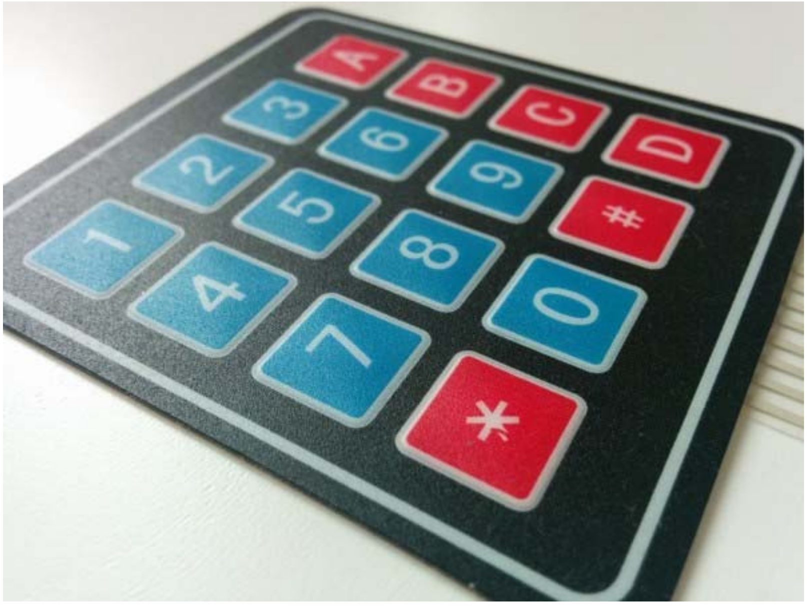
Accelerometer Controlled Car: Start-Up
September 21, 2014



The Sliding Effect Device
September 13, 2014

TAGS

AC mains indicator Arduino Mega 2560 Arduino Uno bridge rectifier external interrupts gesture LED LPC 1768 matrix keypad photocell port manipulation Ultrasonic Sensor XBee



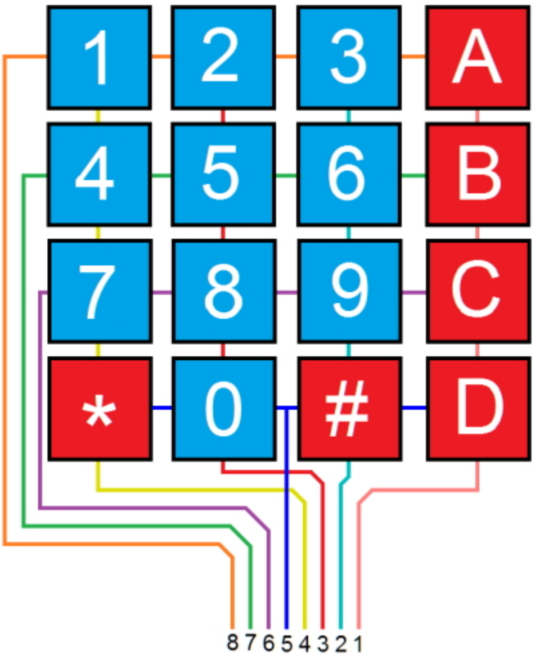
A KEYPAD ACTIVATED INTERRUPT

Posted by: Asem Khattab on September 5, 2014 in Arduino 5 Comments 11,091 views 0

In a previous project, I had an issue. I wanted to be able to activate an interrupt (to temporarily stop what is happening and do something else) through a matrix keypad, so that when I press a specific key the operation stops and then I can use the keypad to update some variables. Unfortunately I couldn't do this and I used a push button instead. But since then I spent time and effort to find a solution and I would like to share it with you here.

How does it Works

Before we start discussing the possible solutions, it is necessary that we know how this matrix keypad works.

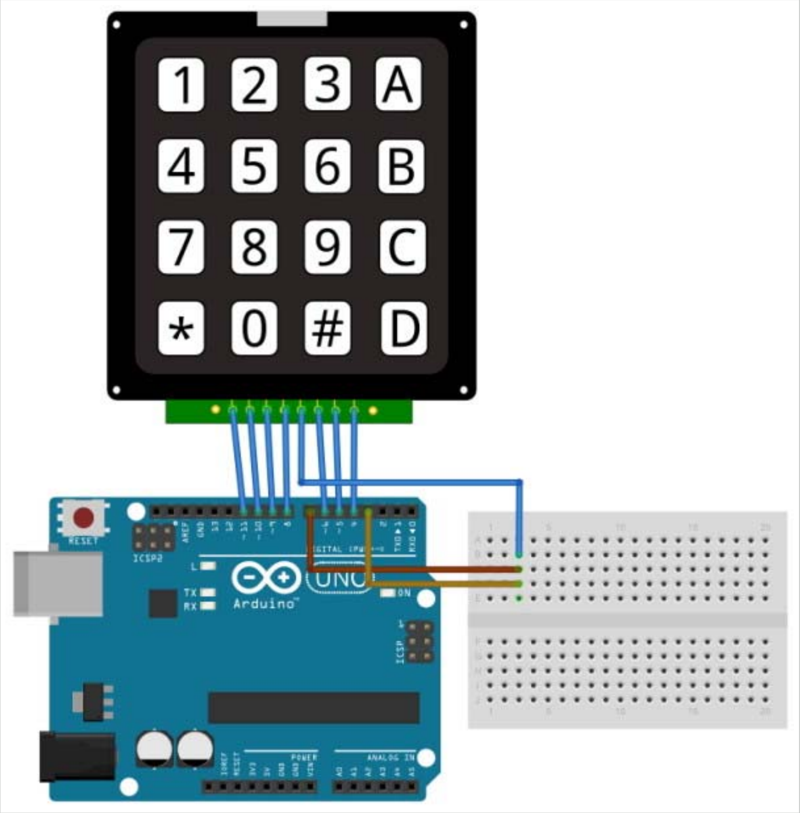


matrix keypad connections – parallax

The keypad is connected by a wire for each column and a wire for each row. according to (Parallax, 2011), the micro-controller detect the pressed key by the following method: the pins connected to the rows are configured as outputs and are made high one by one continuously in a very high speed, and the pins connected to the columns are configured as outputs and scanned continuously. When a key is pressed the high signal goes to the corresponding column. The micro-controller then detects a high signal at that column and determines the row that transmitted this signal.

First Trial

After I knew how this keypad works it seemed obvious for me that if I make all the rows low except one of them, and configure one of the column as input then attach the external interrupt to it and make it triggered when the pin goes high, the interrupt when occur when I press the corresponding key. So I coded a simple program to examine the idea. and connected the hardware like the following diagram:



the code looked like this:

Keypad activated interrupt example 1 Arduino

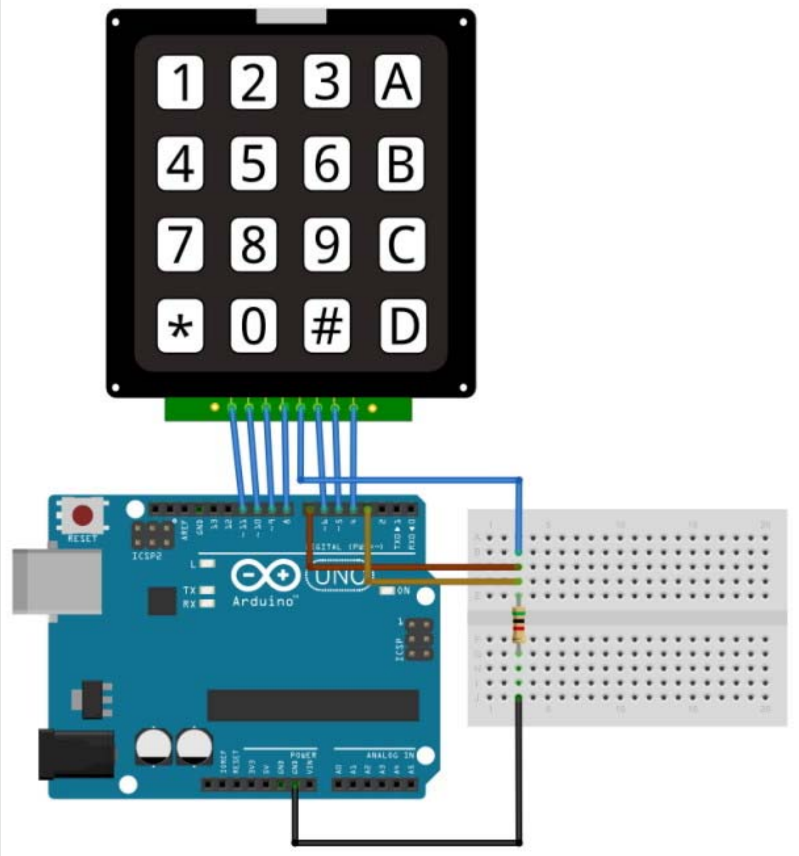
```
//Asem Khattab Aug 2 2014
bool change = false;

//calling the keypad library.
#include <Keypad.h>

const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns

//defining the symbols on the buttons of the keypads.
char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {11, 10, 9, 8};//pins connected to the row pinouts of the
byte colPins[COLS] = {7, 6, 5, 4};  //pins connected to the column pinouts of t

//initialize an instance of class NewKeypad
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COL

//determining the pin connected to an LED.
// Pin 13 has an LED connected on most Arduino boards.
const int led =  13;

void setup() {
  Serial.begin(9600);

  //I want to use the 'A' key, which corresponds to the column 4 (pin 4) and ro
  pinMode(4,INPUT);
  pinMode(11,OUTPUT);
  digitalWrite(11,HIGH);

  // initialize the LED pin as an output:
  pinMode(led, OUTPUT);

  //using the external interrupt of pin 3.
  attachInterrupt(1, getset, HIGH);
}

void loop() {
  //return to these sitting again after using the keypad to input a key.
  pinMode(4,INPUT);
  pinMode(11,OUTPUT);
  digitalWrite(11,HIGH);

  //blink the LED
  digitalWrite(led, HIGH);
  delay(100);
  digitalWrite(led, LOW);
  delay(100);

  if (change == true) getnum();
}

//when 'A' is pressed allow the user to input a key.
void getset() {
  change = true;
}

void getnum() {
  //stop everything till the user press a key.
  char customKey = customKeypad.waitForKey();

  //show the key pressed on the computer through the serial port.
  if (customKey) Serial.println(customKey);

  //remove keypad debounce.
  delay(200);

  //after reading the keypad continue blinking again.
  change = false;
}
```

The program is supposed to blink the LED (L) located at the top left of Arduino Uno till Key ‘A’ is pressed. It should then wait for the user to press another key and display this key on the computer screen, and continue blinking again after that.

This method didn’t work, the interrupt was triggered without even pressing any key. I think this simply because the Arduino keypad library continuously changes the modes and voltages of the columns pins. After some experimentation, I found that some times the voltage at a column pin rises to about 0.4 V, and the micro-controller seems to consider this high enough to trigger the interrupt! while it is stated in this page that Arduino considers the pin high when there is 3 V present at it, So why this happens?

Second Trial

Trying to solve the problem above, I decided to to connect the row and external interrupt pins to the ground through a 5k resistor, so that the connection would be as in the following graph:



I used the same code above.

This method worked, but with some problems. The first problem is that not only the intended key triggers the interrupt, but also the whole column of that key (in this case: column 1 (pin 7)). I discovered also that the part of the code that specify the row and column of the key is not effective and the program behaved the same way without it. so the final code was like the following:

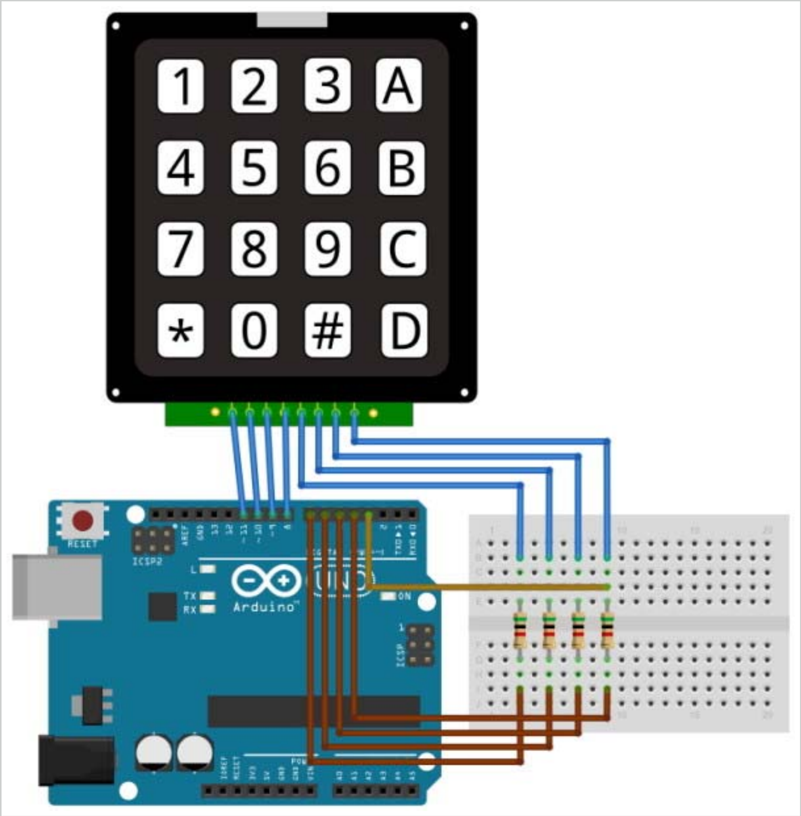
Keypad activated interrupt example 2 Arduino

```
//Asem Khattab Aug 3 2014
//in this example, the first column (starts with '1') is used to activate the i
bool change = false;

//calling the keypad library.
#include <Keypad.h>

const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns

//defining the symbols on the buttons of the keypads.
char hexaKeys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {11, 10, 9, 8};//pins connected to the row pinouts of the
byte colPins[COLS] = {7, 6, 5, 4};  //pins connected to the column pinouts of t

//initialize an instance of class NewKeypad
Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COL

//determining the pin connected to an LED.
// Pin 13 has an LED connected on most Arduino boards.
const int led =  13;

void setup() {
  Serial.begin(9600);

  // initialize the LED pin as an output:
  pinMode(led, OUTPUT);

  //using the external interrupt of pin 3.
  attachInterrupt(1, getset, HIGH);
}

void loop() {
  //blink the LED
  digitalWrite(led, HIGH);
  delay(100);
  digitalWrite(led, LOW);
  delay(100);

  if (change == true) getnum();
}

//when 'A' is pressed allow the user to input a key.
void getset() {
  change = true;
}

void getnum() {
  //stop everything till the user press a key.
  char customKey = customKeypad.waitForKey();

  //show the key pressed on the computer through the serial port.
  if (customKey) Serial.println(customKey);

  //remove keypad debounce.
  delay(200);

  //after reading the keypad continue blinking again.
  change = false;
}
```

The other problem is that when a column other that the first column is used to activate the interrupt, by connecting it to the external interrupt pin then to the ground through a resister, it then inputs a copy of the first column when it is used to input. Lets say we want to use the fourth column (starting with ‘A’), then when I use it to input letter ‘A’, it inputs ‘1’ instead. So again, why this happens? I guess this has something to do with the fact that one of the column is connected in a different way than the others.

Third Trial

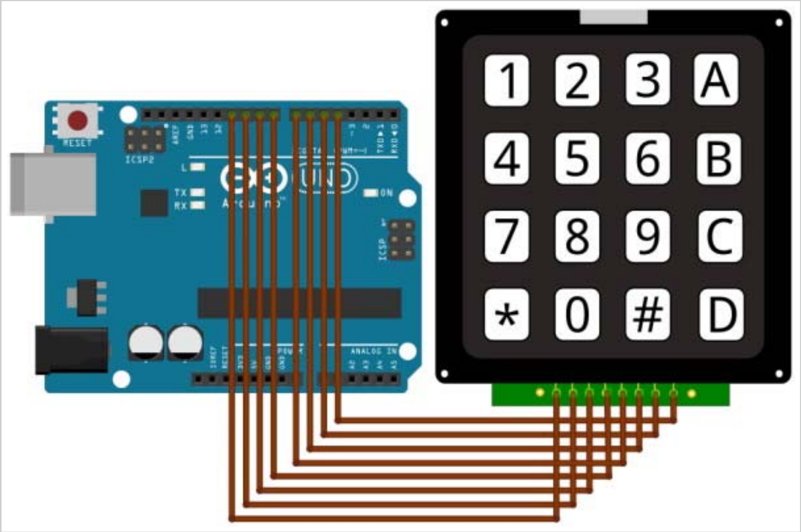
Here I connected the rows to Arduino pins through 5k-ohm resistors, and connect the external interrupt pin to one of them as the following:



I am using key 'A' from the fourth column to activate the interrupt. The code I used is the same as the first one above. The problem of copying the first column is solved here, but the other problem is not. the code that specify the row of the key is ineffective, so the whole column triggers the interrupt. In addition to that, a new problem emerged, the interrupt is sometimes triggered without pressing any key. The problem of the first trial again! I tried a lot to solve the problem by configuring the code and even by modifying the keypad library, but unfortunately, there was no way.

There is always a simple solution

After further searching, I found a feature in the keypad library called "event keypad", which could allow the micro-controller to perform specific actions when a key is pressed. And after some experiments, I could use this feature to be like an interrupt. The connection of the keypad is the usual way the keypad is connected, like the next diagram. Moreover, no external interrupt is used. This is actually more than I ask for!



The code below is built on an example by Alexander Brevig. Many thanks to him.

Keypad activated interrupt example 3 Arduino

```
//Asem Khattab Sep 3 2014

//calling the keypad library.
#include <Keypad.h>

const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns

//defining the symbols on the buttons of the keypads.
char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {11, 10, 9, 8};//pins connected to the row pinouts of the
byte colPins[COLS] = {7, 6, 5, 4};  //pins connected to the column pinouts of t

//initialize an instance of class NewKeypad
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS);

//determining the pin connected to an LED.
// Pin 13 has an LED connected on most Arduino boards.
const int led =  13;

bool change = false;

void setup() {
  Serial.begin(9600);

  // initialize the LED pin as an output:
  pinMode(led, OUTPUT);

  // Add an event listener for this keypad
  keypad.addEventListener(keypadEvent);
}

void loop(){
  //scan the pins of the keypad
  char key = keypad.getKey();

  //blink the LED
  digitalWrite(led, HIGH);
  delay(100);
  digitalWrite(led, LOW);
  delay(100);

  if (change == true) getnum();
}

// Taking care of some special events.
void keypadEvent(KeypadEvent key){
    switch (keypad.getState()){

    //U can use three modes: PRESSED, HOLD or RELEASED.
    //U can also add more than one event with more than one key.
    case PRESSED:
        if (key == 'A') {
            //when 'A' is pressed allow the user to input a key.
            change = true;
        }
        break;
    }
}

void getnum() {
  //stop everything till the user press a key.
  char key = keypad.waitForKey();

  //show the key pressed on the computer through the serial port.
  if (key) Serial.println(key);

  //after reading the keypad continue blinking again.
  change = false;
}
```

This method requires that the keypad pins are "scanned" continuously using the function "getKey()". So, if you have one or more loops, make sure to include this function in each iteration of each loop. This method is highly customizable, you can choose to trigger the action either by pressing, holding or releasing a key or a number of keys.
Finally I found a professional solution.

Tagged with: EXTERNAL INTERRUPTS MATRIX KEYPAD

« Previous:
LEDs Sweeping Effect

Next:
The Sliding Effect Device »

5 COMMENTS

Owis sabry Hussain
September 5, 2014 at 10:10 pm

great ! go ahead....

Reply

paramjeet singh
June 4, 2015 at 11:33 am

actually i have an assignment about arduino calculator. i got an error of no keypad library available.
can you please send me circuit diagram and program coding.

Reply

Asem Khattab
August 3, 2015 at 11:38 pm

you have to download the keypad library and add it to the arduino file. you can find it here http://playground.arduino.cc/Code/Keypad

Reply

tamacdonald@cfl.rr.com
June 29, 2015 at 5:52 am

keypad interrupt works fine using this method i made key=='A' now i want 'A' to call a function
do i need something under loop or setup or getnum() can you help me out with this new to arduino
thank tom if you want my code let me know where to send it to

Reply

Asem Khattab
August 3, 2015 at 11:40 pm

you just need an if-statement that will run when A is pressed. please read this page again and see this: https://www.arduino.cc/en/Reference/If

Reply

LEAVE A REPLY

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

Post Comment

ABOUT AUTHOR

Asem Khattab, an Egyptian and an Electrical Engineering and Computer Science student at Qatar University.
Life it self is nothing but a big project. Here I share with you some of what I am interested in, in a trail to be useful.

FOLLOW ME