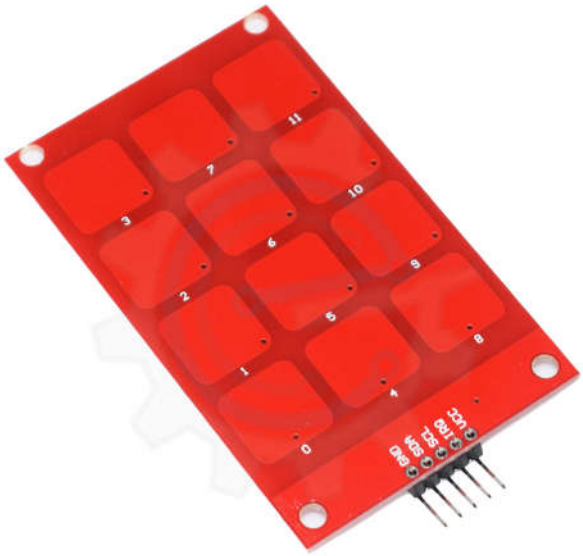
MPR121 Capacitive Touch keypad module



The MPR121 is a capacitive touch sensor controller that makes it very easy to integrate capacitive touch sensing into your project. This signals the MPR121 that something has touched a 'button'. The IC is also capable of driving LEDs or basic GPIO functionality on electrode pins 4 through 11.

Specifications:

Input Voltage: 1.6V to 3.3V DC
Current: 29uA
Communication: I2C
PCB Dimension: 78mm x 49mm

Wiring Connections:

GizduinoV to Capacitive keypad

+5V - VCC
D2 - IRQ
A5/D19 - SCL
A4/D18 - SDA
GND - GND

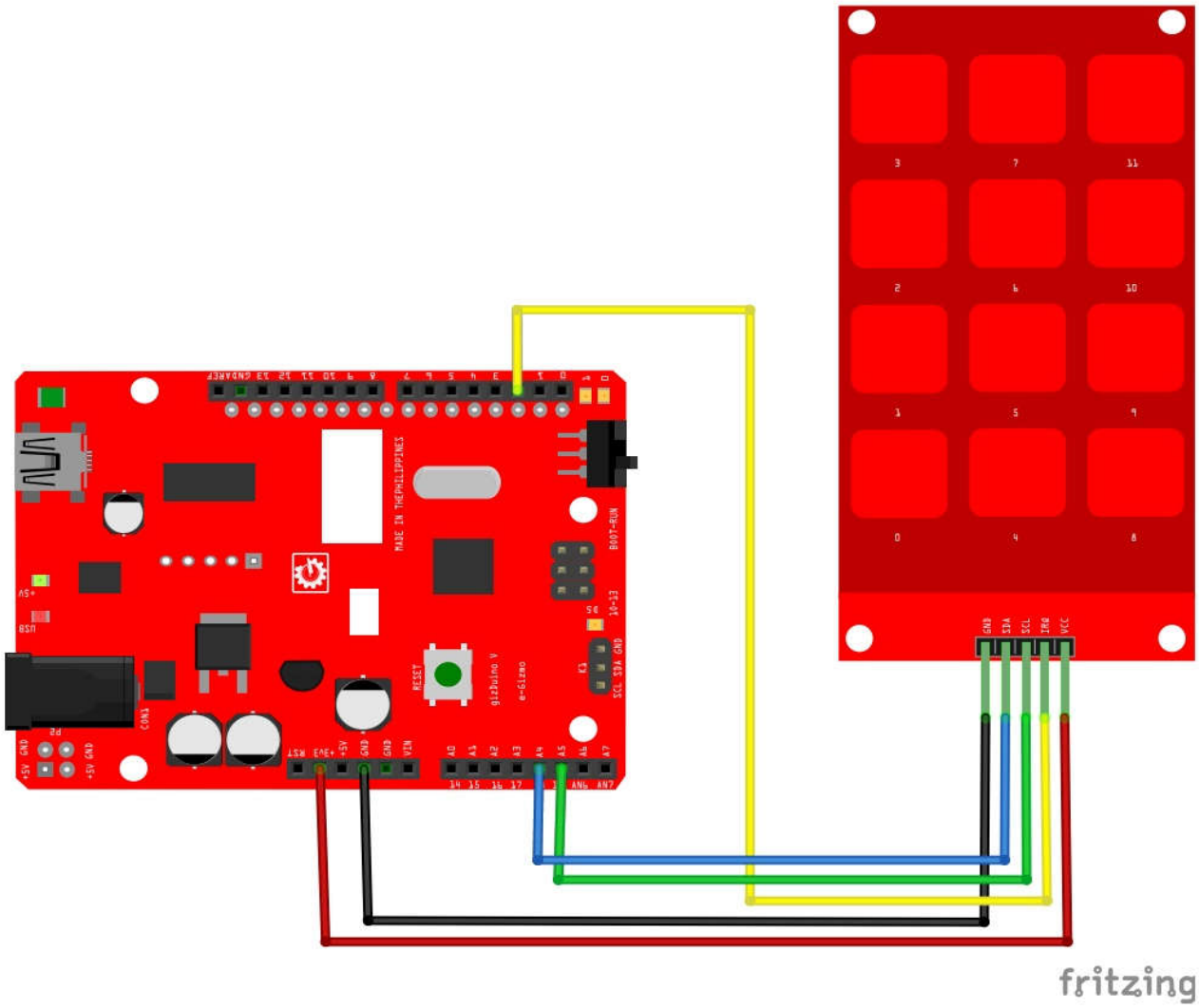


Figure 1. Sample Wiring Diagram with GizDuino V ATmega328P.

```
//****************************************************************//
//            MPR121 Capacitive Keypad Module                     //
//                                                                //
//    This is a sample sketch for Capacitive type of keypad       //
// when the pad is pressed the corresponding number will          //
// print on the serial monitor.                                   //
// Specifications:                                                //
//     Input Voltage: 3.3V or 5V Arduino.                         //
//                                                                //
//    Notes: The Wiring library is not used for I2C, a            //
//         default atmel I2C lib is used. Be sure to keep         //
//         the .h files with the project.                         //
//                                                                //
//  SparkFun Electronics                                          //
//  created on: 6/22/11                                           //
//  license: OSHW 1.0, http://freedomdefined.org/OSHW             //
//                                                                //
// by: Aaron Weiss, based on the MPR121 Keypad                    //
//  Example by                                                    //
// Jim Lindblom                                                   //
// further modified by Jim again! - 6/22/11                       //
//                                                                //
// e-Gizmo Mechatronix Central http://e-gizmo.net                 //
//****************************************************************//

// include the atmel I2C libs
#include "mpr121.h"
#include "i2c.h"

// Match key inputs with electrode numbers
#define ONE 1
#define TWO 2
#define THREE 3
#define FOUR 4
#define FIVE 5
#define SIX 6
#define SEVEN 7
#define EIGHT 8
#define NINE 9

//extras (not connected to button)
#define TEN 10
#define ELEVEN 11
#define ZERO 0

//interupt pin
int irqPin = 2;  // D2

void setup()
```

```
{
  //make sure the interrupt pin is an input and pulled high
  pinMode(irqPin, INPUT);
  digitalWrite(irqPin, HIGH);

  //configure serial out
  Serial.begin(9600);

  // initalize I2C bus. Wiring lib not used.
  i2cInit();

  // initialize mpr121
  mpr121QuickConfig();

  // Create and interrupt to trigger when a button
  // is hit, the IRQ pin goes low, and the function getNumber is run.
  attachInterrupt(0,getNumber,LOW);

  // prints 'Ready...' when you can start hitting numbers
  Serial.println("Ready...");
}

void loop()
{
  //You can put additional code here. The interrupt will run in the backgound.
}

void getNumber()
{
  int touchNumber = 0;
  uint16_t touchstatus;
  char digits;

  touchstatus = getTouchStatus();

  for (int j=0; j<12; j++)  // Check how many electrodes were pressed
  {
    if ((touchstatus & (1<<j)))
      touchNumber++;
  }

  if (touchNumber == 1)
  {
    if (touchstatus & (1<<SEVEN))
    {
      digits = '7';
    }
    else if (touchstatus & (1<<FOUR))
    {
      digits = '4';
    }
```

```
    else if (touchstatus & (1<<ONE))
    {
      digits = '1';
    }
    else if (touchstatus & (1<<EIGHT))
    {
      digits = '8';
    }
    else if (touchstatus & (1<<FIVE))
    {
      digits = '5';
    }
    else if (touchstatus & (1<<TWO))
    {
      digits = '2';
    }
    else if (touchstatus & (1<<NINE))
    {
      digits = '9';
    }
    else if (touchstatus & (1<<SIX))
    {
      digits = '6';
    }
    else if (touchstatus & (1<<THREE))
    {
      digits = '3';
    }
    else if (touchstatus & (1<<TEN))
    {
      digits = 'A';
      tone(BUZZER,550,DUR);
    }
    else if (touchstatus & (1<<ELEVEN))
    {
      digits = 'B';
      tone(BUZZER,650,DUR);
    }
    else if (touchstatus & (1<<ZERO))
    {
      digits = '0';
      tone(BUZZER,600,DUR);
    }
    Serial.println(digits);
  }
  //do nothing if more than one button is pressed,
or if all are released
  else if (touchNumber == 0)
    ;
  else
    ;
}
```

```
/* getTouchStatus() will return a 16-bit value that
relates the
current touched status of each button. The LSB
represents electrodes
0-7 (bit 0 = electrode 0), and the lowest 4 bits of
the MSB
represent electrods 8-11. A 1 means a button is
being touched.*/
int getTouchStatus()
{
  int touch;

  touch = mpr121Read(0x01) << 8;
  touch |= mpr121Read(0x00);

  return touch;
}
```

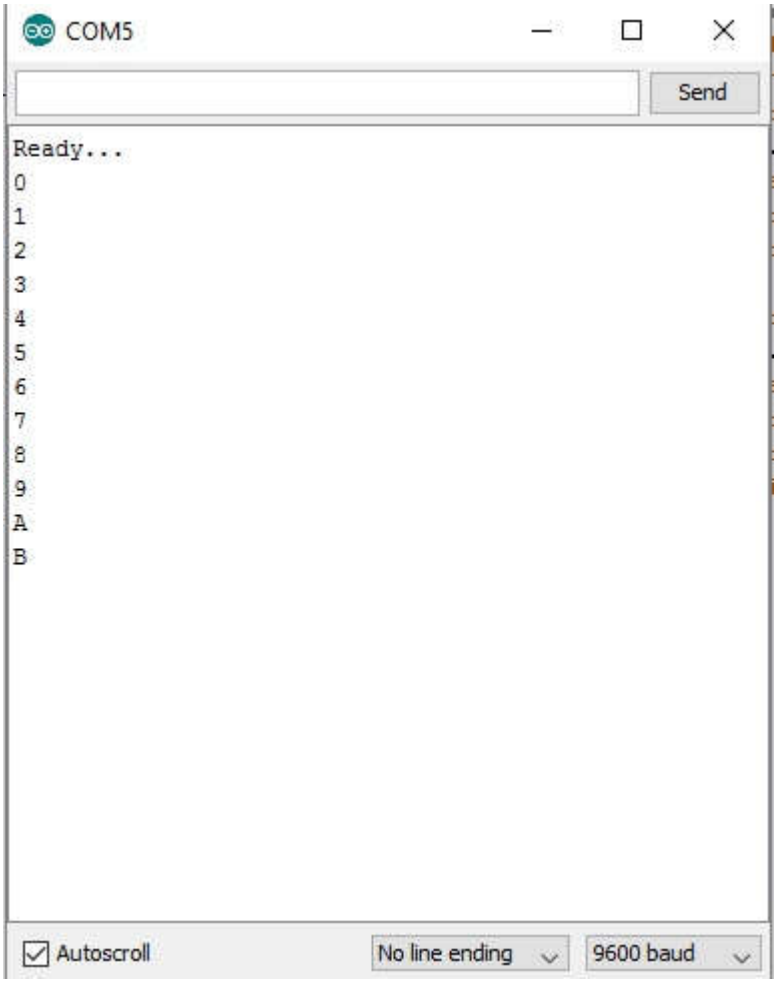


Figure 2. On the Serial monitor you can see the output of Capacitive keypad module.