#include <Wire.h>

#include "Adafruit\_MPR121.h"

#ifndef \_BV

#define \_BV(bit) (1 << (bit))

#endif

#include<SoftwareSerial.h>

// You can have up to 4 on one i2c bus but one is enough for testing!

Adafruit\_MPR121 cap = Adafruit\_MPR121();

SoftwareSerial bt(10,11); //(Rx,Tx)

// Keeps track of the last pins touched

// so we know when buttons are 'released'

uint16\_t lasttouched = 0;

uint16\_t currtouched = 0;

const int ENABLE=7;

int count[100]={0};

void setup() {

bt.begin(9600);

//pinMode.Enable("HIGH");

while (!bt) { // needed to keep leonardo/micro from starting too fast!

delay(10);

}

bt.println("MPR121 Capacitive Touch sensor test");

// Default address is 0x5A, if tied to 3.3V its 0x5B

// If tied to SDA its 0x5C and if SCL then 0x5D

if (!cap.begin(0x5A)) {

bt.println("MPR121 not found, check wiring?");

while (1);

}

bt.println("MPR121 found!");

}

void loop() {

pinMode(ENABLE,"HIGH");

// Get the currently touched pads

currtouched = cap.touched();

for (uint8\_t i=0; i<12; i++) {

// it if \*is\* touched and \*wasnt\* touched before, alert!

if ((currtouched & \_BV(i)) && !(lasttouched & \_BV(i)) ) {

if(count[i]%2!=0)

{

bt.print(char(i+'M'));

count[i]++;

break;

}

else

{

bt.print(char(i+'A'));

count[i]++;

break;

}

}

delay(10);

}

//if it \*was\* touched and now \*isnt\*, alert!

// if (!(currtouched & \_BV(i)) && (lasttouched & \_BV(i)) ) {

// bt.print(char(i+'A')); bt.println(" released");

//}

// reset our state

lasttouched = currtouched;

// comment out this line for detailed data from the sensor!

return;

// debugging info, what

bt.print("\t\t\t\t\t\t\t\t\t\t\t\t\t 0x"); bt.println(cap.touched(), HEX);

bt.print("Filt: ");

for (uint8\_t i=0; i<12; i++) {

bt.print(cap.filteredData(i)); bt.print("\t");

}

bt.println();

bt.print("Base: ");

for (uint8\_t i=0; i<12; i++) {

bt.print(cap.baselineData(i)); bt.print("\t");

}

bt.println();

// put a delay so it isn't overwhelming

delay(100);

}