Supplementary Information: Arduino Sketch Code

// include the library code:

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

// Microfluidic Ring Mixer Program (Gateway Assembly - Button Push for Final Recovery)

// Variable Declaration

const int buttonPin = 6;

int buttonState = 0;

int currentState = 0;

int beenPushed = 0;

int sensorPin = 0; //the analog pin that the TMP36's Vout (sense) pin is connected to

//the resolution is 10 mV / degree centigrade with a

//500 mV offset to allow for negative temperatures

// Setup Function

void setup()

{

// set up the LCD's number of columns and rows:

lcd.begin(16, 2);

pinMode(buttonPin, INPUT);

// valve bank 1 pin designation

pinMode(17, OUTPUT);

pinMode(18, OUTPUT);

pinMode(19, OUTPUT);

pinMode(20, OUTPUT);

pinMode(21, OUTPUT);

pinMode(22, OUTPUT);

pinMode(23, OUTPUT);

pinMode(24, OUTPUT);

//valve bank 2 pin designation

pinMode(25, OUTPUT);

pinMode(26, OUTPUT);

pinMode(27, OUTPUT);

pinMode(28, OUTPUT);

pinMode(29, OUTPUT);

pinMode(30, OUTPUT);

pinMode(31, OUTPUT);

pinMode(32, OUTPUT);

//valve bank 3 pin designation

pinMode(33, OUTPUT);

pinMode(34, OUTPUT);

pinMode(35, OUTPUT);

pinMode(36, OUTPUT);

pinMode(37, OUTPUT);

pinMode(38, OUTPUT);

pinMode(39, OUTPUT);

pinMode(40, OUTPUT);

//valve bank 4 pin designation

pinMode(41, OUTPUT);

pinMode(42, OUTPUT);

pinMode(43, OUTPUT);

pinMode(44, OUTPUT);

pinMode(45, OUTPUT);

pinMode(46, OUTPUT);

pinMode(47, OUTPUT);

pinMode(48, OUTPUT);

// initButton = digitalRead(buttonPin);

}

// Loop Continuously

void loop() {

// Get the State of the button (HIGH when not pushed, LOW when pushed)

buttonState = digitalRead(buttonPin);

// Button is pushed down, set the variable

if (buttonState == LOW) {

beenPushed = 1;

}

// When the current state is 0, set to default configuration

if (currentState == 0) {

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Genetic Circuit Assembly READY");

Ready();

tempSensor();

}

// If the button has been pushed, and is now released, execute code

if (buttonState == HIGH && beenPushed == 1) {

// Reset the push variable

beenPushed = 0;

// Increment the state

currentState = currentState + 1;

// Go do the first routine...

if (currentState == 1) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Latch Device1"); //Prime bank1 valves

Incubate1();

}

// Go do the first routine...

if (currentState == 2) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Latch Device2"); //Prime bank2 valves

Incubate2();

}

// Go do the next routine...

if (currentState == 3) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Block Bank 1"); //Introduce passivating agent Device1

Flow1();

}

// Go do the next routine...

if (currentState == 4) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Incubate Device1"); //Incubate passivating agent Device1

Incubate1();

}

// Go do the next routine...

if (currentState == 5) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Block Bank 2"); //Introduce passivating agent Device2

Flow2();

}

// Go do the next routine...

if (currentState == 6) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Incubate Device 2"); //Incubate passivating agent Device2

Incubate2();

}

// Go do the next routine...

if (currentState == 7) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Purge Device1"); //Flush Device1

Flow1();

}

// Go do the next routine...

if (currentState == 8) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Latch Bank 1");

Incubate1();

}

// Go do the next routine...

if (currentState == 9) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Purge Device2");

Flow2(); //Flush Device2

}

// Go do the next routine...

if (currentState == 10) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Latch Bank 2");

Incubate2();

}

// Go do the next routine...

if (currentState == 11) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Fill Device1");

Fill1();

}

// Go do the next routine...

if (currentState == 12) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Fill Device2");

Fill2();

}

// Go do the next routine...

if (currentState == 13) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Runready Device1");

Runready1();

}

// Go do the next routine...

if (currentState == 14) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Runready Device2");

Runready2();

}

// Go do the next routine...

if (currentState == 15) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Mixing Device1");

Incubate1();

for (int x = 0; x<1000; x++){

Mix1();

}

Incubate1();

}

// Go do the next routine...

if (currentState == 16) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Mixing Device2");

Incubate2();

for (int x = 0; x<1000; x++){

Mix2();

}

Incubate2();

}

// Go do the next routine...

if (currentState == 17) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Incubate Device1");

Incubate1();

}

// Go do the next routine...

if (currentState == 18) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Incubate Device2");

Incubate2();

}

// Go do the next routine...

if (currentState == 19) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Recover Device1 10s");

Flow1();

Incubate1();

}

// Go do the next routine...

if (currentState == 20) {

lcd.clear();

lcd.setCursor(0, 0);

// Print a message to the LCD.

lcd.print("Recover Device2 10s");

Flow2();

Incubate2();

}

}

}

void Ready()

{

// all microfluidic valves open

// valve bank 1

digitalWrite(17, LOW);

digitalWrite(18, LOW);

digitalWrite(19, LOW);

digitalWrite(20, LOW);

digitalWrite(21, LOW);

digitalWrite(22, LOW);

digitalWrite(23, LOW);

digitalWrite(24, LOW);

// valve bank 2

digitalWrite(25, LOW);

digitalWrite(26, LOW);

digitalWrite(27, LOW);

digitalWrite(28, LOW);

digitalWrite(29, LOW);

digitalWrite(30, LOW);

digitalWrite(31, LOW);

digitalWrite(32, LOW);

// valve bank 3

digitalWrite(33, LOW);

digitalWrite(34, LOW);

digitalWrite(35, LOW);

digitalWrite(36, LOW);

digitalWrite(37, LOW);

digitalWrite(38, LOW);

digitalWrite(39, LOW);

digitalWrite(40, LOW);

// valve bank 4

digitalWrite(41, LOW);

digitalWrite(42, LOW);

digitalWrite(43, LOW);

digitalWrite(44, LOW);

digitalWrite(45, LOW);

digitalWrite(46, LOW);

digitalWrite(47, LOW);

digitalWrite(48, LOW);

}

void Incubate1()

{

// Latch valves for bank 1

digitalWrite(17, HIGH);

digitalWrite(18, HIGH);

digitalWrite(19, HIGH);

digitalWrite(20, HIGH);

digitalWrite(21, HIGH);

digitalWrite(22, HIGH);

digitalWrite(23, HIGH);

digitalWrite(24, HIGH);

}

void Incubate2()

{

// Latch valves for bank 2

digitalWrite(25, HIGH);

digitalWrite(26, HIGH);

digitalWrite(27, HIGH);

digitalWrite(28, HIGH);

digitalWrite(29, HIGH);

digitalWrite(30, HIGH);

digitalWrite(31, HIGH);

digitalWrite(32, HIGH);

}

void Flow1()

{

digitalWrite(17, HIGH);

digitalWrite(18, LOW);

digitalWrite(19, LOW);

digitalWrite(20, LOW);

digitalWrite(21, LOW);

digitalWrite(22, LOW);

digitalWrite(23, HIGH);

digitalWrite(24, LOW);

}

void Flow2()

{

digitalWrite(25, HIGH);

digitalWrite(26, LOW);

digitalWrite(27, LOW);

digitalWrite(28, LOW);

digitalWrite(29, LOW);

digitalWrite(30, LOW);

digitalWrite(31, HIGH);

digitalWrite(32, LOW);

}

void Fill1()

{

// Purge ring with air

digitalWrite(17, LOW);

digitalWrite(18, HIGH);

digitalWrite(19, LOW);

digitalWrite(20, LOW);

digitalWrite(21, LOW);

digitalWrite(22, HIGH);

digitalWrite(23, LOW);

digitalWrite(24, HIGH);

}

void Fill2()

{

// Purge ring with air

digitalWrite(25, LOW);

digitalWrite(26, HIGH);

digitalWrite(27, LOW);

digitalWrite(28, LOW);

digitalWrite(29, LOW);

digitalWrite(30, HIGH);

digitalWrite(31, LOW);

digitalWrite(32, HIGH);

}

void Runready1()

{

// green LED to signal block done and OK to run protocol

digitalWrite(17, HIGH);

digitalWrite(18, HIGH);

digitalWrite(19, LOW);

digitalWrite(20, LOW);

digitalWrite(21, HIGH);

digitalWrite(22, HIGH);

digitalWrite(23, HIGH);

digitalWrite(24, HIGH);

}

void Runready2()

{

// green LED to signal block done and OK to run protocol

digitalWrite(25, HIGH);

digitalWrite(26, HIGH);

digitalWrite(27, LOW);

digitalWrite(28, LOW);

digitalWrite(29, HIGH);

digitalWrite(30, HIGH);

digitalWrite(31, HIGH);

digitalWrite(32, HIGH);

}

void Mix1()

{

// peristaltic pump routine (25 Hz)

digitalWrite(17, HIGH);

digitalWrite(18, HIGH);

digitalWrite(22, LOW);

digitalWrite(23, HIGH);

digitalWrite(24, HIGH);

digitalWrite(19, LOW);

digitalWrite(20, LOW);

digitalWrite(21, LOW);

delay(40);

digitalWrite(19, HIGH);

digitalWrite(20, LOW);

digitalWrite(21, LOW);

delay(40);

digitalWrite(19, HIGH);

digitalWrite(20, HIGH);

digitalWrite(21, LOW);

delay(40);

digitalWrite(19, HIGH);

digitalWrite(20, HIGH);

digitalWrite(21, HIGH);

delay(40);

digitalWrite(19, LOW);

digitalWrite(20, HIGH);

digitalWrite(21, HIGH);

delay(40);

digitalWrite(19, LOW);

digitalWrite(20, LOW);

digitalWrite(21, HIGH);

delay(40);

}

void Mix2()

{

// peristaltic pump routine (25 Hz)

digitalWrite(25, HIGH);

digitalWrite(26, HIGH);

digitalWrite(30, LOW);

digitalWrite(31, HIGH);

digitalWrite(32, HIGH);

digitalWrite(27, LOW);

digitalWrite(28, LOW);

digitalWrite(29, LOW);

delay(40);

digitalWrite(19, HIGH);

digitalWrite(20, LOW);

digitalWrite(21, LOW);

delay(40);

digitalWrite(27, HIGH);

digitalWrite(28, HIGH);

digitalWrite(29, LOW);

delay(40);

digitalWrite(27, HIGH);

digitalWrite(28, HIGH);

digitalWrite(29, HIGH);

delay(40);

digitalWrite(27, LOW);

digitalWrite(28, HIGH);

digitalWrite(29, HIGH);

delay(40);

digitalWrite(27, LOW);

digitalWrite(28, LOW);

digitalWrite(29, HIGH);

delay(40);

}

void Incubate1()

{

// Latch valves for bank 1

digitalWrite(17, HIGH);

digitalWrite(18, HIGH);

digitalWrite(19, HIGH);

digitalWrite(20, HIGH);

digitalWrite(21, HIGH);

digitalWrite(22, HIGH);

digitalWrite(23, HIGH);

digitalWrite(24, HIGH);

}

void Incubate2()

{

// Latch valves for bank 1

digitalWrite(25, HIGH);

digitalWrite(26, HIGH);

digitalWrite(27, HIGH);

digitalWrite(28, HIGH);

digitalWrite(29, HIGH);

digitalWrite(30, HIGH);

digitalWrite(31, HIGH);

digitalWrite(32, HIGH);

}

void Recover1()

{

//moves the ligated product to the chip outlet for recovery

digitalWrite(17, HIGH);

digitalWrite(18, LOW);

digitalWrite(19, LOW);

digitalWrite(20, LOW);

digitalWrite(21, LOW);

digitalWrite(22, LOW);

digitalWrite(23, HIGH);

digitalWrite(24, LOW);

delay(10000);

}

void Recover2()

{

//moves the ligated product to the chip outlet for recovery

digitalWrite(25, HIGH);

digitalWrite(26, LOW);

digitalWrite(27, LOW);

digitalWrite(28, LOW);

digitalWrite(29, LOW);

digitalWrite(30, LOW);

digitalWrite(31, HIGH);

digitalWrite(32, LOW);

delay(10000);

}

void tempSensor()

{

//getting the voltage reading from the temperature sensor

int reading = analogRead(sensorPin);

// converting that reading to voltage, for 3.3v arduino use 3.3

float voltage = reading \* 5.0;

voltage /= 1024.0;

float temperatureC = (voltage - 0.5) \* 100 ; //converting from 10 mv per degree wit 500 mV offset

//to degrees ((volatge - 500mV) times 100)

lcd.setCursor(12, 1);

// Print a message to the LCD.

lcd.print(temperatureC,0);

lcd.setCursor(14, 1);

lcd.print("C");

}