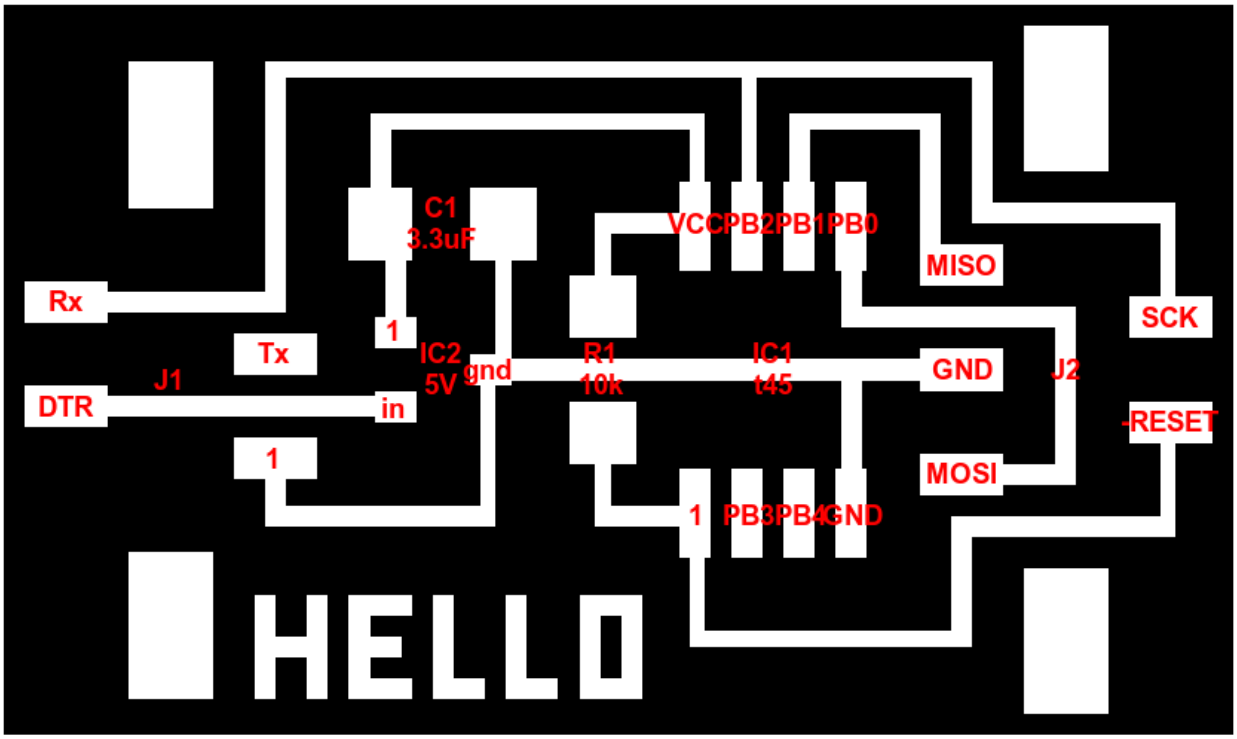
Serial board



Components

Symbol	Component	Package	Notes
C1	Capacitor 3.3µF	SMD1206	
IC1	Microcontroller ATTiny45	SOIC	Pin labeled 1 corresponds to small dot on the microcontroller.
IC2	Regulator 5V 100mA	SOT23	
J1	4 pin MTA connector	MTA	RS232 communications port
J2	5 pin MTA connector	MTA	ISP programming interface
R1	Resistor 10kΩ	SMD1206	

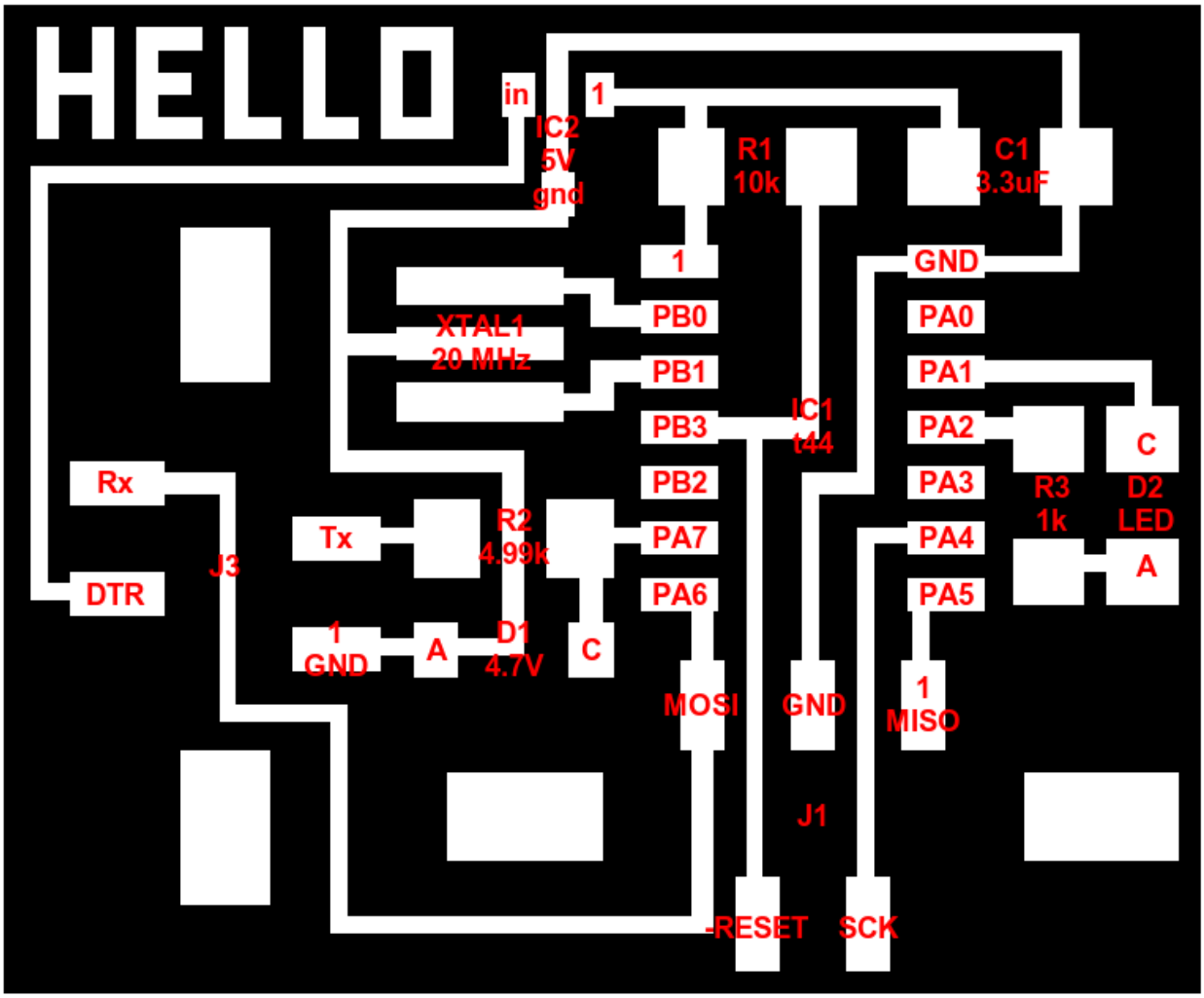
Description

A simple circuit that uses the RS-232 (serial) communications standard to send the 8 bit ASCII texta string "Hello World!" over and over at 9600 bits per second.

Files

- hello.serial.45.cad – circuit schematic
- hello.serial.45.asm - assembler language code

Serial communications port (echo circuit)



Components

Symbol	Component	Package	Notes
C1	Capacitor 3.3µF	SMD1206	
D1	Diode 4.7V	SMD1206	Check orientation. C = Cathode, labeled with green line on component.
D2	Light Emitting Diode (LED)	SMD1206	Check orientation. C = Cathode, labeled with green line on component.
IC1	Microcontroller ATTiny44	SOIC	Pin labeled 1 corresponds to small dot on the microcontroller.
IC2	Regulator 5V 100mA	SOT23	
J1	5 pin MTA connector	MTA	ISP programming interface
J3	4 pin MTA connector	MTA	RS232 communications port
R1	Resistor 10kΩ	SMD1206	
R2	Resistor 4.99kΩ	SMD1206	

R3	Resistor 1kΩ	SMD1206	
XTAL1	Oscillator 20 MHz		

Description

A slightly more advanced circuit that uses RS-232 (serial) communications to listen for text and “echos” it back at 9600 bits per second.

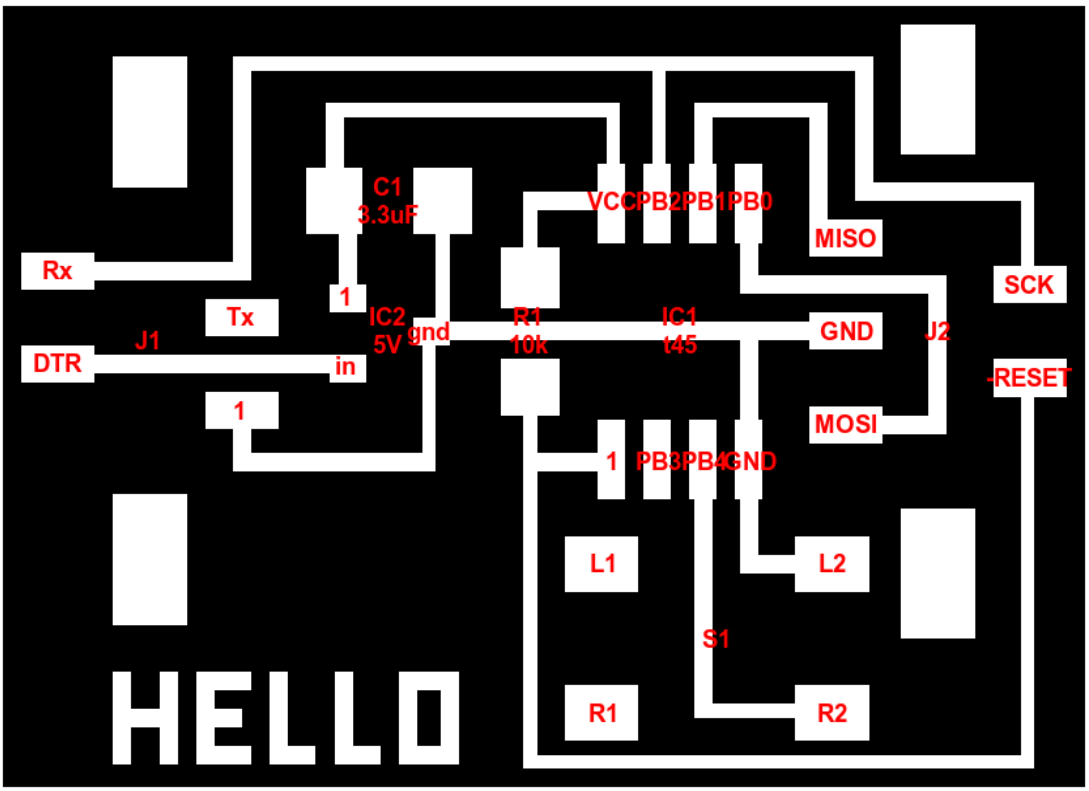
This circuit is easy to change to do more advanced things. As a programming exercise, try turning it into a postfix calculator that listens for symbols on the serial port and sends results back.

If you want something more challenging, try implementing a rudimentary web server!

Files

- hello.echo.44.cad – circuit schematic, ATTiny44
- hello.echo.44.asm - assembler language code, ATTiny44

Simple button circuit



Components

Symbol	Component	Package	Notes
C1	Capacitor 3.3µF	SMD1206	
IC1	Microcontroller ATTiny45	SOIC	Pin labeled 1 corresponds to small dot on the microcontroller.
IC2	Regulator 5V 100mA	SOT23	
J2	5 pin MTA connector	MTA	ISP programming interface
J1	4 pin MTA connector	MTA	RS232 communications port
R1	Resistor 10kΩ	SMD1206	
S1	Pushbutton	PLCC	

Description

Similar to the first serial 'hello' board, but has a button that can be used to wait for pushbutton events. The included program sends a message through the serial port when the button is pressed.

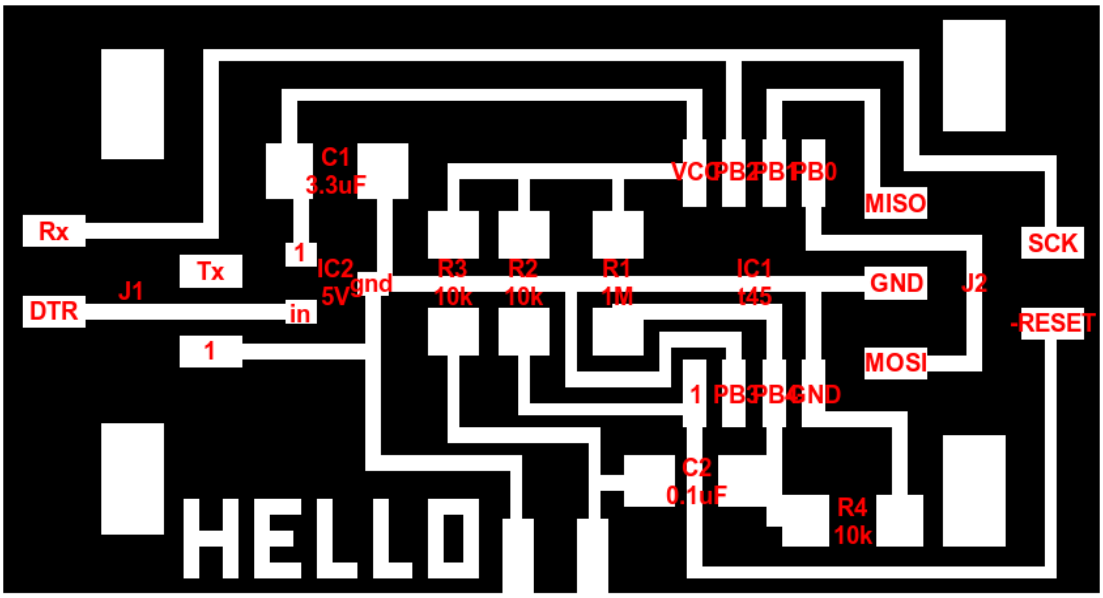
As a programming exercise, you could try sending how many times the button has been pressed, or how many seconds since it was last pressed.

If you want to be more advanced, try changing it into a Morse key that transmits the corresponding letters over the Serial port.

Files

- hello.button.45.cad - circuit schematic
- hello.button.45.asm - assembler language code

Microphone



Components

Symbol	Component	Package	Notes
C1	Capacitor 3.3µF	SMD1206	
IC1	Microcontroller ATTiny45	SOIC	Pin labeled 1 corresponds to small dot on the microcontroller.
IC2	Regulator 5V 100mA	SOT23	
J2	5 pin MTA connector	MTA	ISP programming interface
J1	4 pin MTA connector	MTA	RS232 communications port
R1	Resistor 1MΩ	SMD1206	
C2	Capacitor 0.1µF	SMD1206	
R2	Resistor 10kΩ	SMD1206	
R3	Resistor 10kΩ	SMD1206	
R4	Resistor 10kΩ	SMD1206	
Unlabeled	Microphone		Connects to two pins at bottom of board; check orientation.

Description

A circuit that uses a condensing microphone to capture sound waves and sends their values over the serial port. The included Python program then draws these on the screen.

As a programming exercise, try extending the Python program to save the captured audio as a PCM (.wav) file. Check the recording.

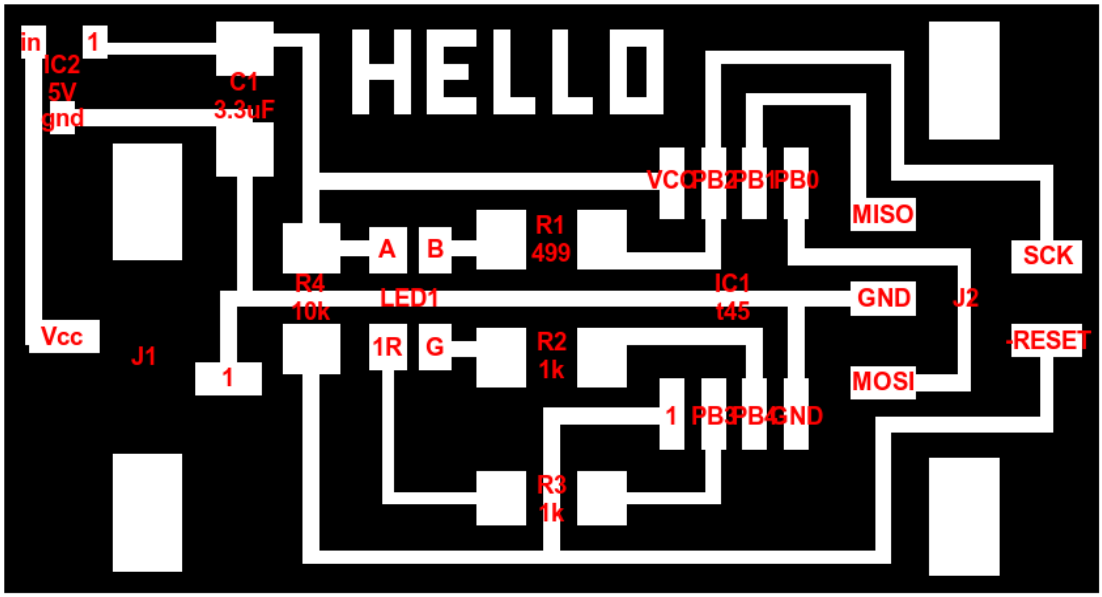
How is the quality? How do you think it can be improved?

Files

- hello.mic.45.cad - circuit schematic

- hello.mic.45.asm - assembler language code
- hello.mic.45.py - Python program that shows the measured sound waveform in real time

Hello.RGB.45.cad



Components

Symbol	Component	Package	Notes
C1	Capacitor 3.3µF	SMD1206	
IC1	Microcontroller ATTiny45	SOIC	Pin labeled 1 corresponds to small dot on the microcontroller.
IC2	Regulator 5V 100mA	SOT23	
J2	5 pin MTA connector	MTA	ISP programming interface
J1	2 pin MTA connector	MTA	Power, 5V DC.
R1	Resistor 499Ω	SMD1206	
R2	Resistor 1kΩ	SMD1206	
R3	Resistor 1kΩ	SMD1206	
R4	Resistor 10kΩ	SMD1206	
LED1	Light Emitting Diode, RGB		No dot inside the LED on the anode.

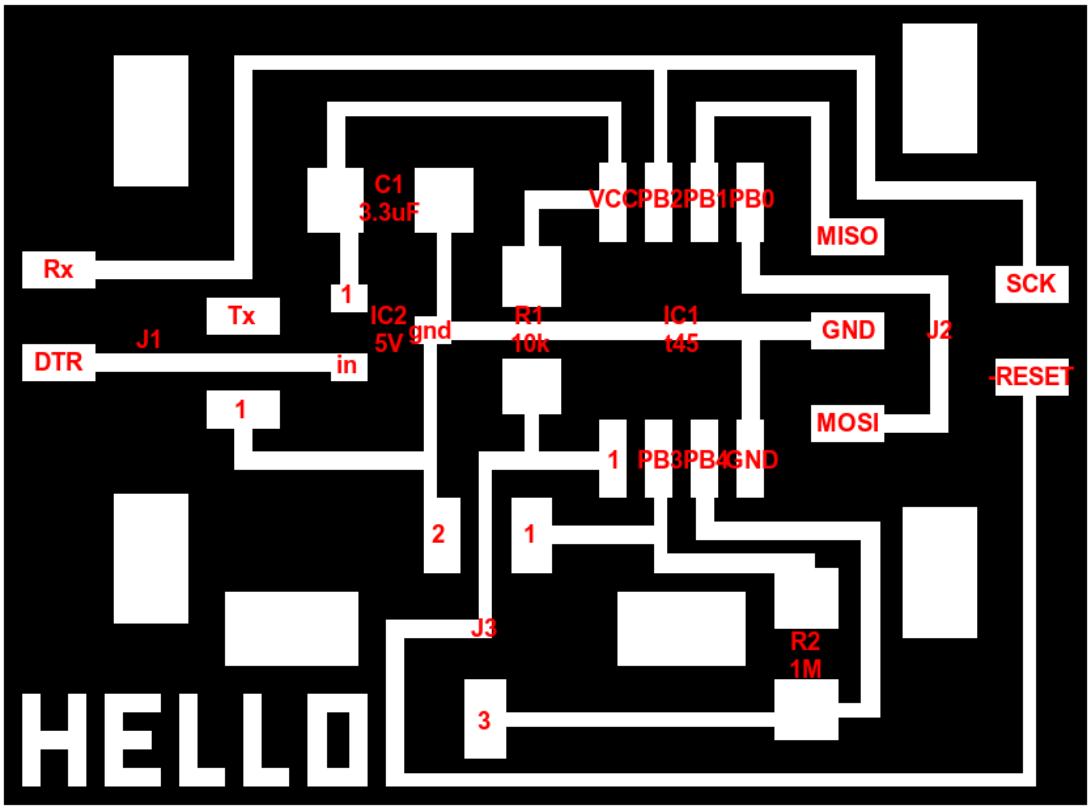
Description

Circuit with an RGB LED.

Files

- hello.RGB.45.cad - circuit schematic
- hello.RGB.45.asm - assembler language code – program that cycles through colors.

Hello.step.45.cad



Components

Symbol	Component	Package	Notes
C1	Capacitor 3.3µF	SMD1206	
IC1	Microcontroller ATTiny45	SOIC	Pin labeled 1 corresponds to small dot on the microcontroller.
IC2	Regulator 5V 100mA	SOT23	
J1	5 pin MTA connector	MTA	ISP programming interface
J2	4 pin MTA connector	MTA	Communications port, RS-232
R1	Resistor 10kΩ	SMD1206	
R2	Resistor 1MΩ	SMD1206	
J3	3 pin MTA connector	MTA	Sensor connector

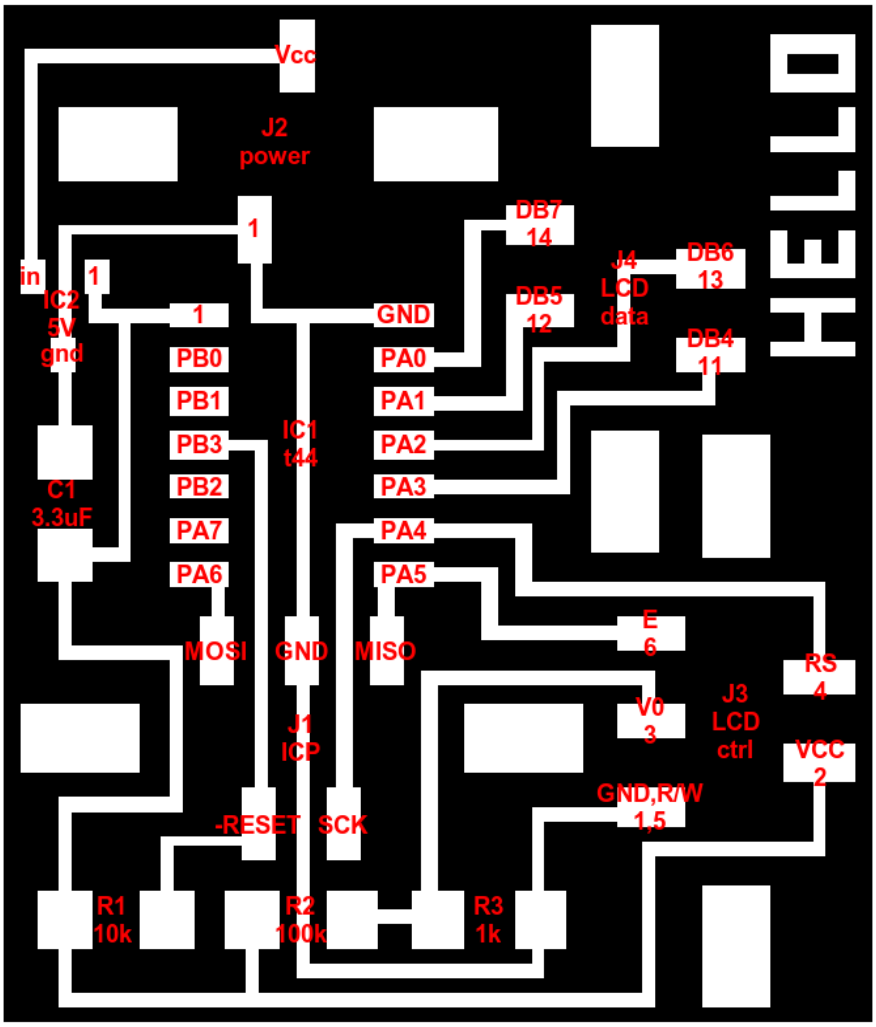
Description

Circuit that measures any kind of step response, such as resistance, capacitance, span, location or distance. Connect different sensors or components to the circuit to measure different things.

Files

- hello.step.45.cad - circuit schematic
- hello.step.45.asm - assembler language code
- hello.step.45.py - Python program that listens on the serial port and displays measured values.

Hello.LCD.44.cad



Components

Symbol	Component	Package	Notes
C1	Capacitor 3.3µF	SMD1206	
IC1	Microcontroller ATTiny44	SOIC	Pin labeled 1 corresponds to small dot on the microcontroller.
IC2	Regulator 5V 100mA	SOT23	
J1	5 pin MTA connector	MTA	ISP programming interface
J2	2 pin MTA connector	MTA	Power, 5V DC.
R1	Resistor 10kΩ	SMD1206	
R2	Resistor 100kΩ	SMD1206	
J3	5 pin MTA connector	MTA	LCD control port
J4	4 pin MTA connector	MTA	LCD data port
R3	Resistor 1kΩ	SMD1206	

Description

Controls a standard 13 pin LCD screen in 4 bit mode. Pins on the data port connect to pins 0-3 on

the LCD screen. The control port is then connected to the control pins on the LCD: ground, R/W, VCC, V0, E and RS, in that order. The numbers on the schematic show which pins on the LCD to connect to. Note that ground (GND) should be jumpered between the two ground pins 1 and 5.

Files

- hello.LCD.44.cad - circuit schematic
- hello.LCD.44.asm - assembler language code