Ladder Logic GUI for Arduino

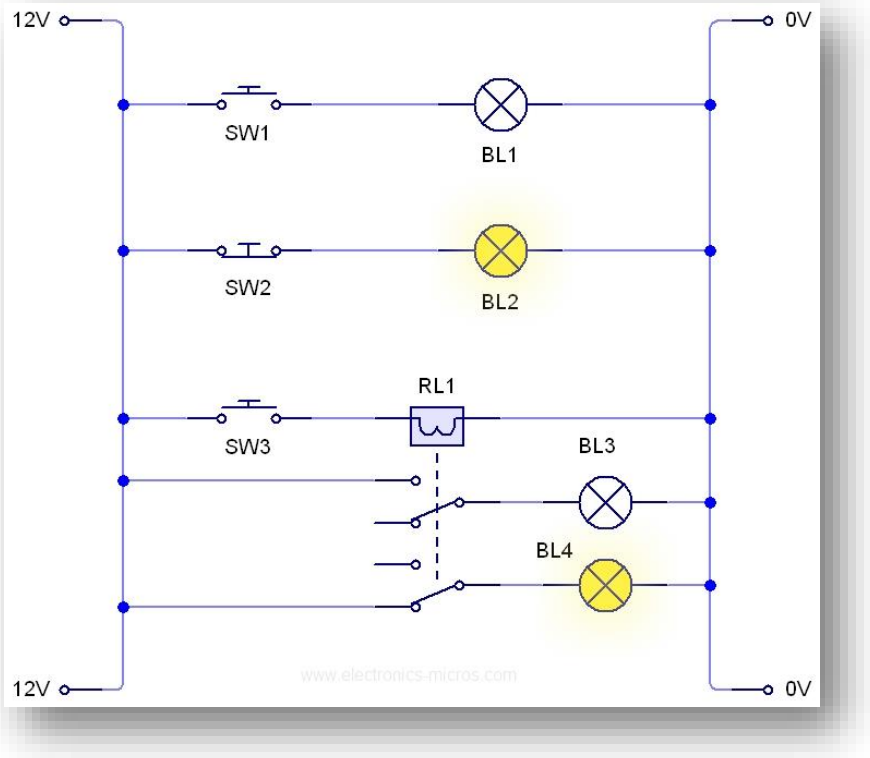
The scope of this project is to create a tool where the user can edit Ladder Logic programs and upload them to Arduino.

Electrical engineers and automation specialists are used to this kind of notation when using programmable logic controllers (PLC).

The name 'ladder diagram' comes from the superficial resemblance to a physical ladder, with vertical power rails at each side, and horizontal circuit branches called rungs connected between the rails. More complex ladder diagrams have a series of rungs, each of which represents a separate circuit.

Ladder diagrams are an adaptation of an earlier technology called relay logic, in which switches and relays are used to control industrial circuits.

Here's an example of a ladder logic diagram:



Notice the positive power rail at the left, and negative at the right. Switches SW1 and SW2 are push-to-make and push-to-break types, causing their associated lamps to be lit when the switches are pressed or released, respectively. Switch SW3 is connected to relay coil RL1 and the changeover relay contacts are

then linked to lamps BL3 and BL4. The relay contacts are arranged so that only one lamp is lit at any time.

The Arduino system, instead, is programmed by using a C program that gets compiled and uploaded by a toolkit made by Atmel called AVRDude.

Here's a screenshot of the Arduino code

```
#include <plcLib.h>

/* Programmable Logic Controller Library for the Arduino and Compatibles

   Digital Input Output - Single bit I/O in normal and inverted forms

   Connections:
   Input - switch connected to input X0 (Arduino pin A0 / Tinkerkit pin I0)
   Input - switch connected to input X1 (Arduino pin A1 / Tinkerkit pin I1)
   Input - switch connected to input X2 (Arduino pin A2 / Tinkerkit pin I2)
   Input - switch connected to input X3 (Arduino pin A3 / Tinkerkit pin I3)
   Output - LED connected to output Y0 (Arduino pin 3 / Tinkerkit O5)
   Output - LED connected to output Y1 (Arduino pin 5 / Tinkerkit O4)
   Output - LED connected to output Y2 (Arduino pin 6 / Tinkerkit O3)
   Output - LED connected to output Y3 (Arduino pin 9 / Tinkerkit O2)

   Software and Documentation:
   http://www.electronics-micros.com/software-hardware/plclib-arduino/

*/

void setup() {
  setupPLC();  // Setup inputs and outputs
}

void loop() {
  in(X0);      // Read Input 0
  out(Y0);     // Send to Output 0

  inNot(X1);   // Read Input 1 (inverted)
  out(Y1);     // Send to Output 1

  in(X2);      // Read Input 2
  outNot(Y2);  // Send to Output 2 (inverted)

  inNot(X3);   // Read Input 3 (inverted) and send to Output 3 (inverted)
  outNot(Y3);  // (The double negative cancels out)
}
```

In this example, the ladder logic commands are spelled out in C.

We need a graphical tool that draws the ladder diagram and writes the C code included in the void loop() function in the example above.

HIghlights:

1 – The tool has to be written in C++ for portability (using frameworks like QT, gtk, etc.) but the target of choice (at least initially) will be Windows.

2 – The tool will just launch the avrdude and related command line utilities, it will NOT need to do any special compiler processing or real time debugging.

3 – As a bonus, the tool might have to validate the logic diagram possibly according to the IEC 61131-3 standard (http://en.wikipedia.org/wiki/IEC_61131-3) . But it truth these are the only allowable instructions for the Arduino:

Single Bit Digital Input / Output

Command	Description	Example
in(input);	Reads a digital input	in(X0);
out(output);	Outputs to a digital output	out(Y0);
inNot(input);	Reads an inverted digital input	inNot(X0);
outNot(output);	Outputs an inverted signal to a digital output	out(Y0);

Combinational Logic

Command	Description	Example
andBit(input);	Logical AND with a digital input	andBit(X0);
orBit(input);	Logical OR with a digital input	orBit(X0);
xorBit(input);	Logical XOR with a digital input	xorBit(X0);
andNotBit(input);	Logical AND with an inverted digital input	andNotBit(X0);
orNotBit(input);	Logical OR with an inverted digital input	orNotBit(X0);

Analogue Signals

Command	Description	Example
inAnalog(input);	Reads an analogue input	inAnalog(input);
outPWM(output);	Outputs a PWM waveform	outPWM(output);
outServo(output);	Outputs to a servo	outServo(output);

Latches

Command	Description	Example
latch(latch_output, reset_input);	Latches the previous digital input value	in(X0); latch(Y0, X1);
latchKey(set_key, reset_key, output);	Latches a digital output based on keypad entry	latchKey('1', '2', Y0);

Timers

Command	Description	Example

timerOn(timer_variable, delay_ms);	Produces a delayed output after an input is enabled	in(X0);
		timerOn(TIMER0, 2000);
		out(Y0);

timerOff(timer_variable, delay_ms);	Delays turning an output off after an input is removed	in(X0);
		timerOff(TIMER0, 2000);
		out(Y0);

timerPulse(low_elapsed_timer, low_ms,
high_elapsed_timer, high_ms);
	Creates a repeating pulse waveform, if enabled	in(X0);
		timerPulse(AUX1, AUX, AUX3, AUX4);
		out(Y0);