

Plcshield

Find this online on <http://automation.cirotec.mx/plcshield> (<http://automation.cirotec.mx/plcshield>)

Hardware

The shield contains 6 analog inputs (0-5V), 6 digital inputs (12/24V) with isolated grounds (1, 2 and 3 share an isolated ground, and 4, 5 and 6 share another isolated ground), and 6 relay outputs.

There is also an LCD screen, which is controlled via I2C.

Besides the LCD screen, there are 3 buttons, which can be used to control program flow.

There is a reset button, and also a connector to program the Arduino.

The board needs an Arduino Ethernet to run.

The schematic can be found under the PCB folder


Software

Plcshield.h is a library we developed for this project.

Installation

Download this repo and move all the contents of plcshield/libraries to your Arduino libraries folder (eg. C:\Users\ExampleUser\Documents\Arduino\libraries)

Public functions

 **void** plc_setup()

Description: This function must be called prior to any other call to this library. It initializes the inputs and outputs (outputs are initialized to NC). LCD is initialized and it displays a sample text.

Parameters: Nothing

Returns: Nothing

```
uint8_t plc_digitalWrite(uint8_t digital_output, uint8_t level)
```

Description: Digital write to a PLC output.

Parameters:

digital_output – Digital output number [1 .. 6]

level – Desired logical level in output [1 .. 0]

Returns: 0 if digital_output is inside the range 1 to 6, -1 otherwise.

```
uint8_t plc_digitalRead(uint8_t digital_input)
```

Description: Digital read from a PLC input.

Parameters:

digital_input – Digital input number [1 .. 6]

Returns: Logic level in input [1 .. 0] or -1 if the argument given is not in the range 1 to 6.

```
uint8_t plc_buttonRead(uint8_t button)
```

Description: Read a button from the PLC.

Parameters:

button – Button number [1 .. 3]

Returns: Logic level in button [1 .. 0] or -1 if the argument given is not in the range 1 to 3.

```
int plc_analogRead(uint8_t analog_input)
```

Description: Read an analog input from the PLC.

Parameters:

analog_input – Analog input number [1 .. 6]

Returns: 10 bit ADC value [0 .. 1024] or -1 if the argument given is not in the range 1 to 6.

Example

```
#import "plcshield.h"

void setup(){
  plc_setup(); // Setup PLC library
}
```

```

void loop(){
  int a = plc_digitalRead(1); // Read PLC digital input 1
  int b = plc_analogRead(5); // Read PLC analog input 5
  plc_digitalWrite(1,a); // Write PLC digital output 1 to the level read in a
}

```

Using LCD

The library used for LCD control is (<https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library> (<https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>)). For advanced instructions in the LCD (register modification, control commands, etc) refer to their documentation.

Inside our library, there is an instance of LCD called `plc_ldc`, which is initialized during `plcSetup()`.

All you need to do is call `plc_lcd.exampleFunction()`, where `exampleFunction` is any of the functions outlined here:

LCD Functions

`plc_lcd.clear()` - clear LCD screen

`plc_lcd.home()` - Set cursor position to 0

`plc_lcd.noDisplay()` - Turn the display off `plc_lcd.display()` - Turn the display on

`plc_lcd.noCursor()` - Turns the underline cursor off `plc_lcd.cursor()` - Turns the underline cursor on

`plc_lcd.noBlink()` - Turn off blinking cursor `plc_lcd.blink()` - Turn on blinking cursor

`plc_lcd.noBacklight()` - Turn off backlight `plc_lcd.backlight()` - Turn on backlight

`plc_lcd.setCursor(uint8_t col, uint8_t row)` - Set cursor in a determined cell (in the 16 x 2 display)

Example

```

#import "plcshield.h"
void setup()
{
  plc_setup(); // Initialize PLC Library

  plc_lcd.clear(); // Clear the screen

  plc_lcd.noBlink(); // Cursor does not blink

```

```
plc_lcd.noCursor(); // Hide cursor

plc_lcd.setCursor(0,0); // Set the cursor in Column 0, Row 0 of the LCD

plc_lcd.print("Cirotec"); // Print this text where the cursor is

plc_lcd.setCursor(0,1); // Set the cursor in Column 0, Row 1 of the LCD

plc_lcd.print("Automation"); // Print this text where the cursor is
}

void loop()
{
}
```