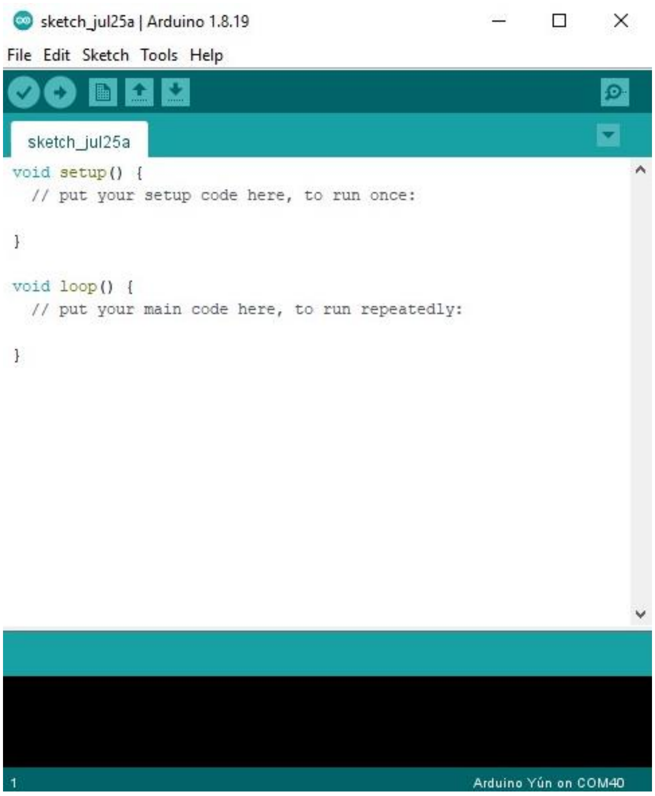
Power Meter Monitor

Business and Mission-

Critical Solutions Provider

PMM06 Integration with Arduino IDE

User Manual



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Document: Guidelines
Document version: 1.0
Date: July 2022

Contents

1. INTRODUCTION

This Document is a fully descriptive guidelines for integrating PMM06 series with Arduino IDE. Providing the operator with the needed information in terms of instructions and screen layout allowing

1.1 Description

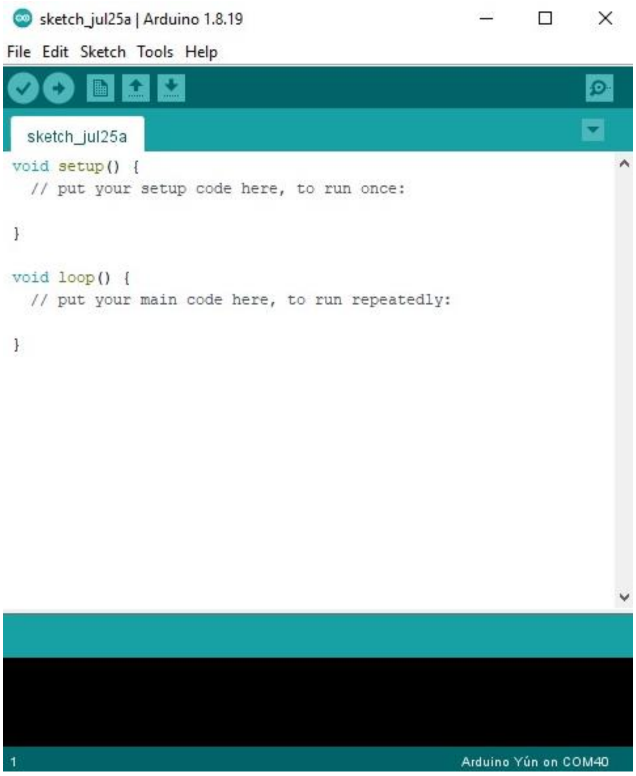
The open-source Arduino Software (IDE) is the most used IDE for Arduino and makes it easy to write code and upload it to PMM06 series. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. In order to work on Arduino (IDE) Make sure you install the libraries first as stated in the instruction manual.

1.2 List of Compatible devices

- PMM0612
- PMM0620
- PMM0625
- PMM0626
- PMM0627
- PMM0628
- PMM0630
- PMM0631
- PMM0632
- PMM0635
- PMM0636
- PMM0638
- PMM0639

2. INTEGRATION GUIDELINES

1. Open open-source Arduino Software (IDE), the following main layout will be displayed as shown below.



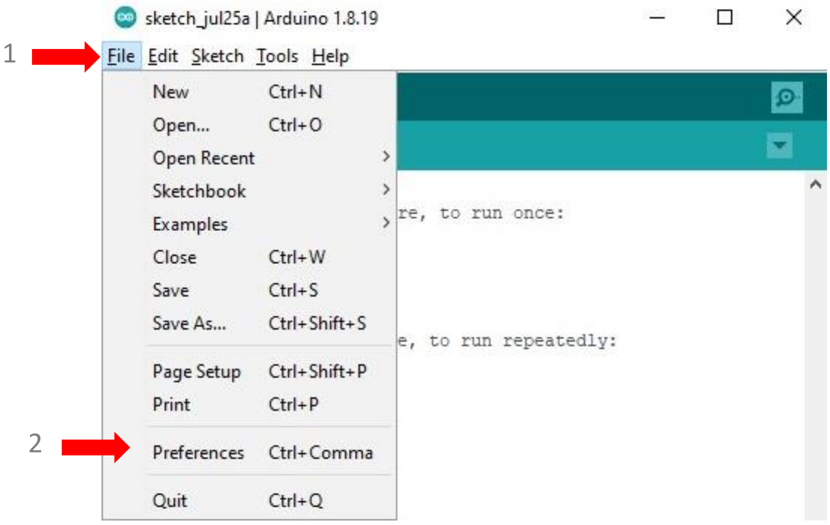
```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

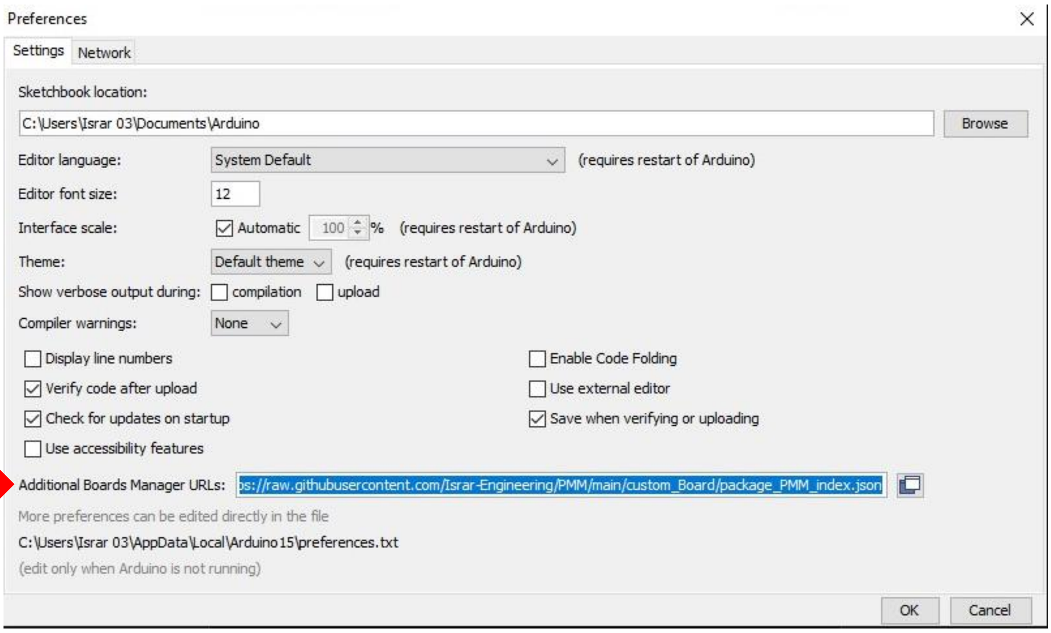
2. Click on "File" in the main tabs and choose "Preferences" or press on the keyboard "Ctrl+Comma".



3. The preferences window will be displayed as shown below, the user can edit the settings as needed.
4. At the "Additional Boards Manager URLs" place the following link in order to download the PMM boards.

PMM Boards Definition on Arduino Link: https://raw.githubusercontent.com/Israr-Engineering/PMM/main/custom_Board/package_PMM_index.json

5. Click on “OK”.



Preferences

Settings | Network

Sketchbook location:

C:\Users\Israr 03\Documents\Arduino

Editor language: System Default (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: □ compilation □ upload

Compiler warnings: None

□ Display line numbers

☑ Verify code after upload

☑ Check for updates on startup

□ Use accessibility features

□ Enable Code Folding

□ Use external editor

☑ Save when verifying or uploading

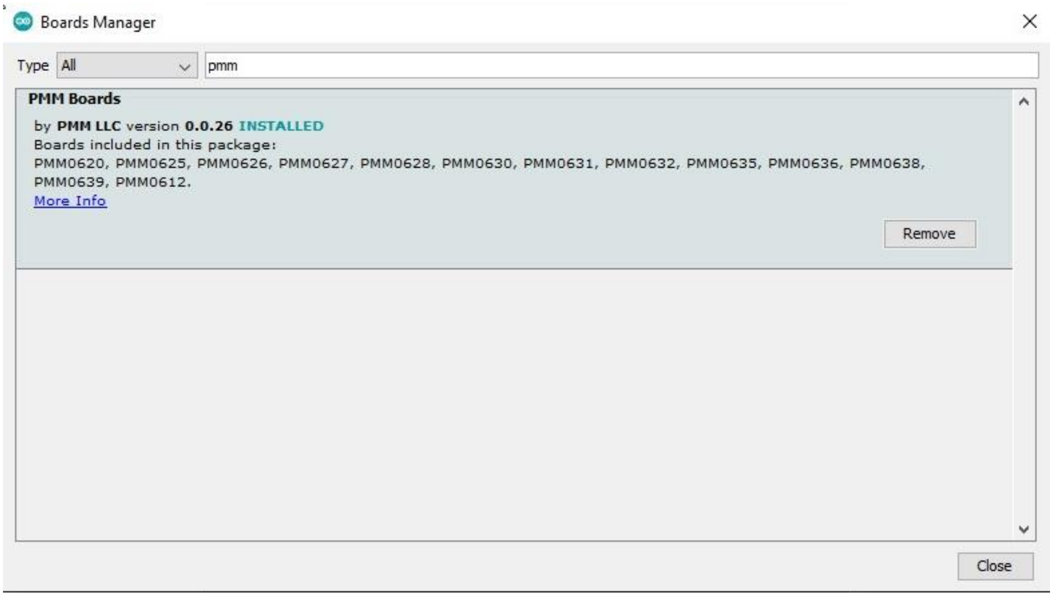
Additional Boards Manager URLs: ps://raw.githubusercontent.com/Israr-Engineering/PMM/main/custom_Board/package_PMM_index.json

More preferences can be edited directly in the file

C:\Users\Israr 03\AppData\Local\Arduino15\preferences.txt

(edit only when Arduino is not running)

6. The “Boards Manager” displays all the PMM board that are included in the downloaded package.



Boards Manager

Type All | pmm

PMM Boards

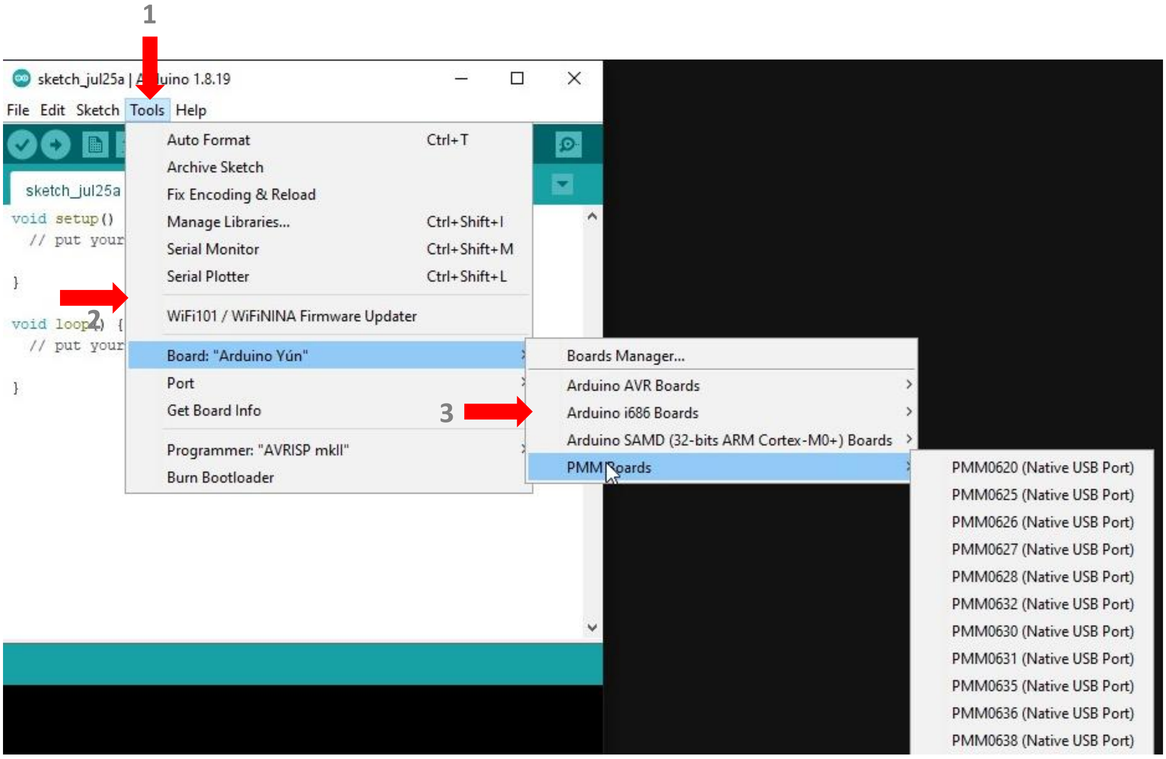
by **PMM LLC** version **0.0.26** INSTALLED

Boards included in this package:

PMM0620, PMM0625, PMM0626, PMM0627, PMM0628, PMM0630, PMM0631, PMM0632, PMM0635, PMM0636, PMM0638, PMM0639, PMM0612.

More Info

7. To choose a specific board to update its code; click on “Tools” in the main tab.
8. Click on “Board”.
9. Click on “PMM Boards”.
10. All the downloaded PMM board will be displayed in the list, choose the desired one.

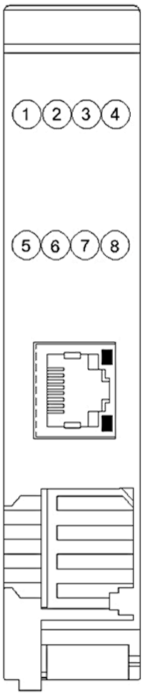

1
sketch_jul25a | Arduino 1.8.19
File Edit Sketch Tools Help
sketch_jul25a
void setup() {
// put your
}
void loop() {
// put your
}
2
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Yún"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
3
Boards Manager...
Arduino AVR Boards
Arduino i686 Boards
Arduino SAMD (32-bits ARM Cortex-M0+) Boards
PMM Boards
PMM0620 (Native USB Port)
PMM0625 (Native USB Port)
PMM0626 (Native USB Port)
PMM0627 (Native USB Port)
PMM0628 (Native USB Port)
PMM0632 (Native USB Port)
PMM0630 (Native USB Port)
PMM0631 (Native USB Port)
PMM0635 (Native USB Port)
PMM0636 (Native USB Port)
PMM0638 (Native USB Port)

3. PMM0625 INTEGRATION with Arduino IDE

This section is full descriptive of the instructions related to connecting PMM0625 and editing the code using Arduino IDE. PMM0625-T is a reliable digital output module with 8 (80VDC) transistor isolated channels. The module sends digital signals from the CPU to the field actuators controlling their status ON/Off.

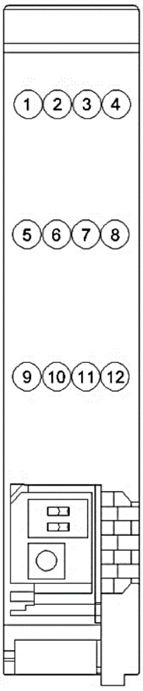
3.1 Pin Assignments

TOP VIEW



1. D+ (RS485)
2. D- (RS485)
3. GND
4. Master command
5. V DC+ (9-56)
6. V DC- (9-56)
7. Earth
8. Earth

BOTTOM VIEW



1. Digital output 01
2. Digital output 02
3. Digital output 03
4. Digital output 04
5. Digital output 05
6. Digital output 06
7. Digital output 07
8. Digital output 08
9. VCC
10. COM1
11. VCC
12. COM 2

3.2 Hardware Connections

Connecting Power

PMM0625-T has two power supply options 10-60 VDC (10-48 VAC), the user has to connect the positive power line (+) to pin no.5 in the top view and the negative line (-) to pin no.6 as illustrated in the pin's assignments.

Note: the power is protected against overvoltage and reverse polarity in case of wrong connection.

Connecting Serial Device

The unit's serial port is located on the top panel. If you are connecting an RS485 multidrop network with multiple devices, note the following:

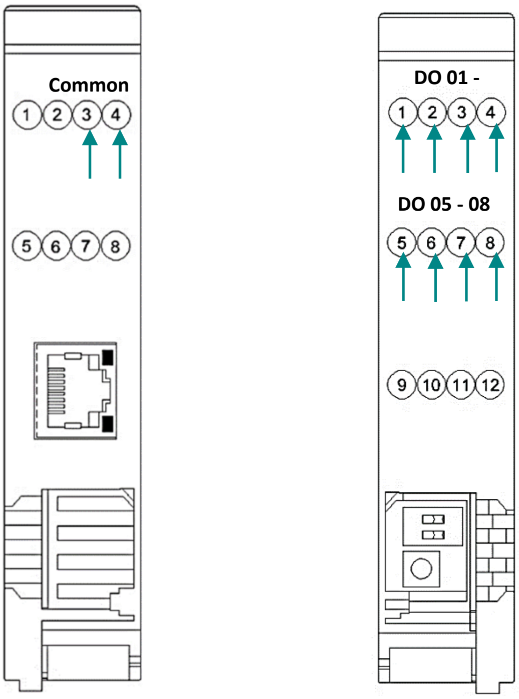
- All devices that are connected to a single serial port must use the same protocol (i.e., either Modbus RTU or Modbus ASCII).
- Connect the D+ with pin no.1 and D- with pin no.2 and Earth with pin no.4 as illustrated in the pin's assignments to complete the connection successfully.
- Turn on the dip switch to have 120 Ω termination resistor between the D+ and D- lines. Refer hardware configuration section.

Connecting to a Host or the Network

There is a 10/100 Ethernet port at the module's top panel. This port is used to connect the module with a host or Ethernet network.

Connecting Digital Output

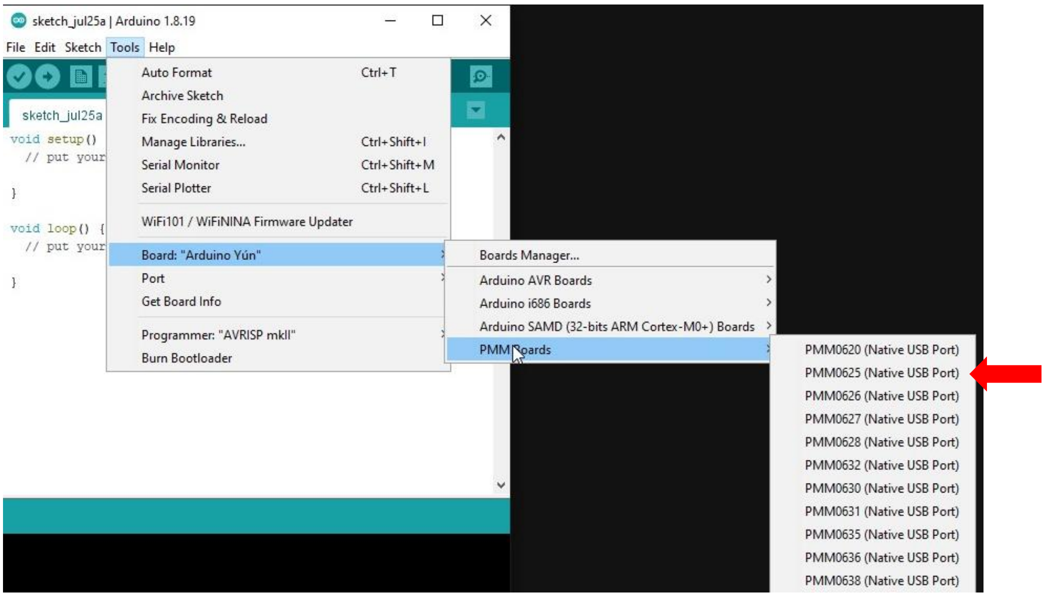
- Connect the signal line with one of the twelve digital output pins on the bottom view (01-08) and the common line with pin no.10 or 12.



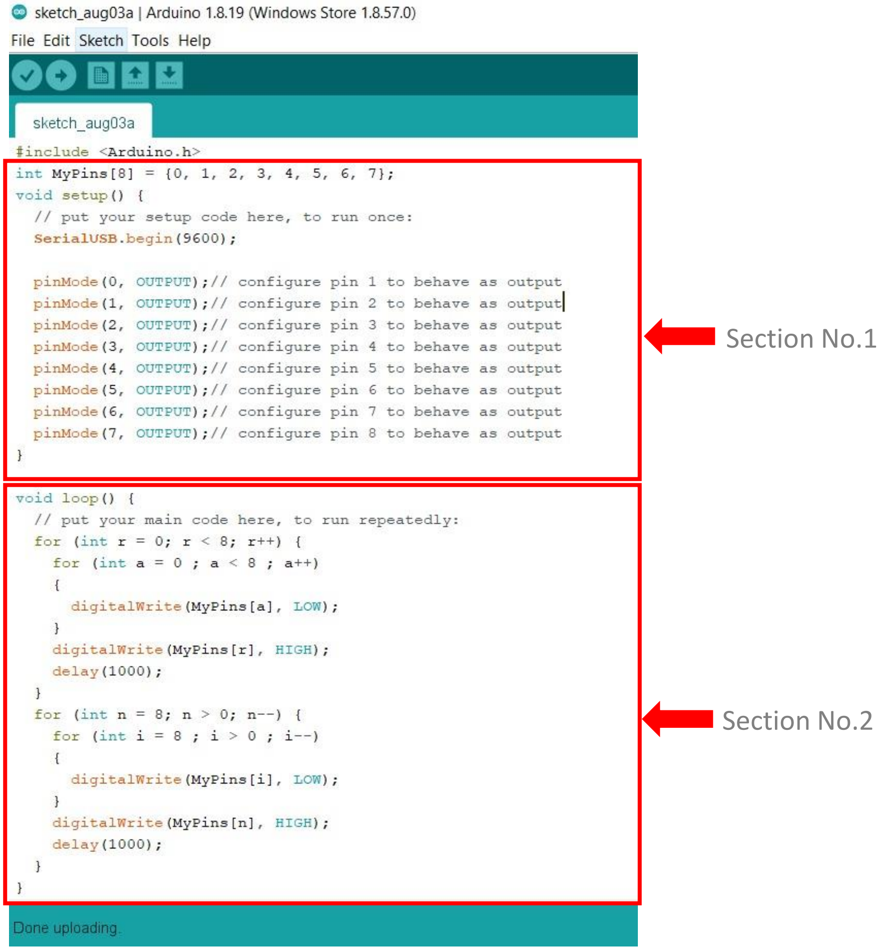
Connecting the USB

Connect the USB to the device through the USB port in the front panel (Micro-USB type), and connect the other side with personal computer (PC). Once the USB is connected correctly between the device and PC, the user can start the integration as explained in the Integration Guidelines:

- Click on “PMM0625” to edit the code.


sketch_jul25a | Arduino 1.8.19
File Edit Sketch Tools Help
sketch_jul25a
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "Arduino Yún"
Port
Get Board Info
Programmer: "AVRISP mkII"
Burn Bootloader
Boards Manager...
Arduino AVR Boards
Arduino i686 Boards
Arduino SAMD (32-bits ARM Cortex-M0+) Boards
PMM Boards
PMM0620 (Native USB Port)
PMM0625 (Native USB Port)
PMM0626 (Native USB Port)
PMM0627 (Native USB Port)
PMM0628 (Native USB Port)
PMM0632 (Native USB Port)
PMM0630 (Native USB Port)
PMM0631 (Native USB Port)
PMM0635 (Native USB Port)
PMM0636 (Native USB Port)
PMM0638 (Native USB Port)

- After choosing PMM0625 board, the following code will be displayed.
- The user can identify the output pins in section No.1 as shown in the figure.
- The user can set the digital write to be Low or High from section No.2 as well as specifying the response time.


sketch_aug03a | Arduino 1.8.19 (Windows Store 1.8.57.0)
File Edit Sketch Tools Help
sketch_aug03a
Section No.1
Section No.2

```
#include <Arduino.h>
int MyPins[8] = {0, 1, 2, 3, 4, 5, 6, 7};
void setup() {
  // put your setup code here, to run once:
  SerialUSB.begin(9600);

  pinMode(0, OUTPUT);// configure pin 1 to behave as output
  pinMode(1, OUTPUT);// configure pin 2 to behave as output
  pinMode(2, OUTPUT);// configure pin 3 to behave as output
  pinMode(3, OUTPUT);// configure pin 4 to behave as output
  pinMode(4, OUTPUT);// configure pin 5 to behave as output
  pinMode(5, OUTPUT);// configure pin 6 to behave as output
  pinMode(6, OUTPUT);// configure pin 7 to behave as output
  pinMode(7, OUTPUT);// configure pin 8 to behave as output
}
void loop() {
  // put your main code here, to run repeatedly:
  for (int r = 0; r < 8; r++) {
    for (int a = 0 ; a < 8 ; a++)
    {
      digitalWrite(MyPins[a], LOW);
    }
    digitalWrite(MyPins[r], HIGH);
    delay(1000);
  }
  for (int n = 8; n > 0; n--) {
    for (int i = 8 ; i > 0 ; i--)
    {
      digitalWrite(MyPins[i], LOW);
    }
    digitalWrite(MyPins[n], HIGH);
    delay(1000);
  }
}
```

Done uploading.