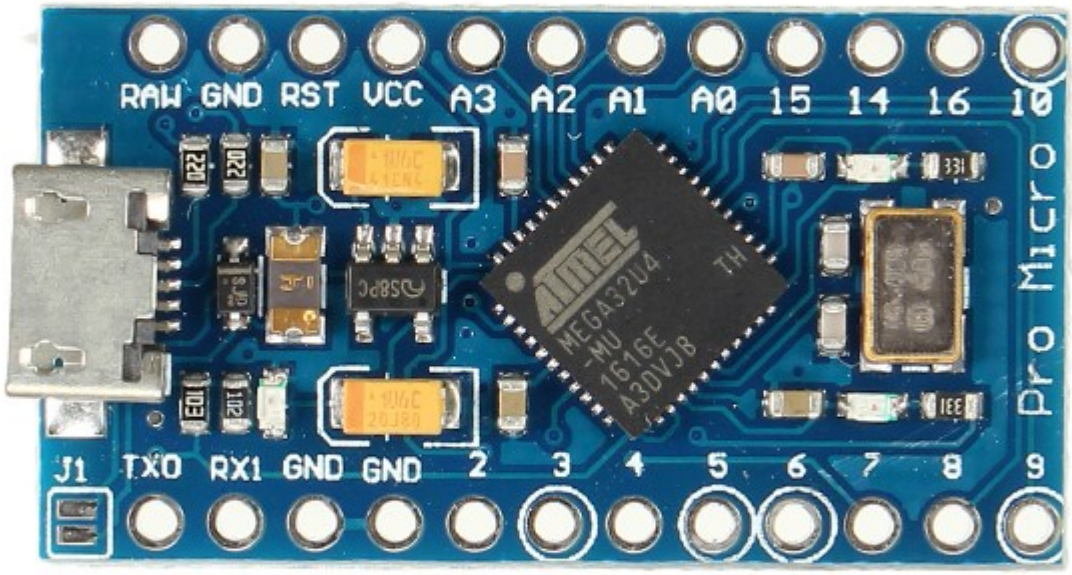
Arduino Controller



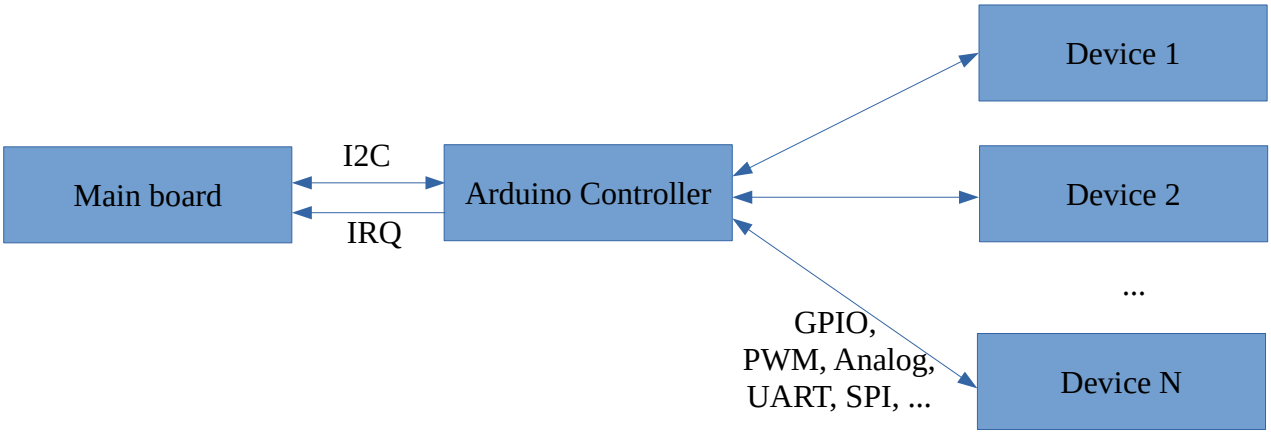
Release Date: 11/10/2019
Author: Sebastien Malissard

Table of Contents

1 Presentation

1.1 Introduction

Arduino Controller allows a main board (Raspberry Pi Arduino, or any micro with I2C interface) to control multiple devices with an Arduino (or any other micro-controller). The following image shows the overall architecture.



To communicate with the Arduino Controller, the I2C interface and an optional IRQ are used. I2C is the main protocol use to communicate. All commands and status packet are sent or received from the I2C. The IRQ allow the Arduino Controller to tell the main board that the status of a device has changed.

On the other hand, the Arduino Controller is connected to several devices that can be controlled using different methods: a simple GPIO, PWM, Analog output, UART, SPI… The list of the devices available is specific to the implementation of the Arduino Controller, see example in Annexe A with Robot HMI.

1.2 I2C configuration

1.2.1 Specification

The Main board is the I2C master and the Arduino Controller is a slave with an address between 0x61 and 0x69.

The default I2C speed of Arduino Controller is 400 kbit/s (Fast Mode).

1.2.2 Communication

To control or get information, the main board sent Command packet and the Arduino Controller sent Status packet.

2 Command packet

[Device] [ID] [Command] [Parameter 1] … [Parameter N]

[Device]	The device family to send the command (led, gpio, servo, switch, …)
[ID]	Id of the device to send the command
[Command]	Command device to execute
[Parameter 1 … N]	Associated command parameter

A command packet without parameter is a read request and the status packet will return the value of each parameter.

When a command is tag with RO (Read-only), only a read request can be done.

3 Status packet

Return the status of the last command received:
[Device] [ID] [Status]

In case of a read request, the status of the command and the value of each parameter is returned:
[Device] [ID] [Status] [Value 1] … [Value N]

List of the status.

Value	**Name**	**Comment**
0x00	OK	No error
0x01	ERROR	Device error
0x02	INVALID_COMMAND	Command not supported
0x03	INVALID_ID	Device id not supported
0x04	INVALID_DEVICE	Device not supported

4 Devices

The following table list all the devices available.

Device	**Name**	**Comment**
0x00	SYSTEM	Internal instruction to the Arduino Controller
0x01	GPIO	Simple gpio input or output
0x02	LED	LED monocolor
0x03	LED_RGB	LED RGB
0x04	SWITCH	Simple push switch or switch
0x05	SERVO	Servo controlled by PWM
0x06	AX12	Servo dynamixel AX12

0x07	DISPLAY_ 7SEG_4DIG	7 segments display with 4 digits
0x08	MOTOR	Drive a CC motor
...		
0xFF	IRQ	Reserved for IRQ register

4.1 LED

The following table list the commands supported by the LED device.

Command	Name	Parameters	Comment
0x01	POWER	[power]	power boolean: 0 = off other value = on
0x02	INTENSITY	[intensity]	intensity [0 – 255]: 0 = 0% 255 = 100%
0x03	BLINK	[period MSB] [period LSB] [duty cycle]	period in ms [0-65535] duty cycle [0-255]: 0 = 0% 255 = 100%

4.2 LED_RGB

The following table list the commands supported by the LED_RGB device.

Command	Name	Parameters	Comment
0x01	POWER	[power]	power boolean: 0 = off other value = on
0x02	COLOR	[color]	Theoretical color (wikipedia): 0 = none 1 = red 2 = yellow 3 = green 4 = cyan 5 = blue 6 = magenta 7 = white
0x03	PWM	[red] [green] [blue]	pwm red, green and blue [0 – 255]: 0 = 0% 255 = 100%

0x04	BLINK	[period MSB] [period LSB] [duty cycle]	period in ms [0-65535] duty cycle [0-255]: 0 = 0% 255 = 100%

4.3 BUTTON

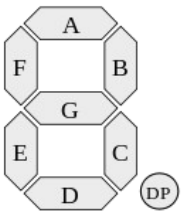
The following table list the commands supported by the BUTTON device.

Command	Name	Parameters	Comment
0x01 (RO)	STATE	[state]	state boolean: 0 = unpress other value = press

SWITCH are IRQ compatible.

4.4 DISPLAY_7SEG_4DIG

The following table list the commands supported by the DISPLAY_7SEG_4DIG device.

Command	Name	Parameters	Comment
0x01	POWER	[power]	power boolean: 0 = off other value = on
0x02	INTENSITY	[intensity]	intensity [0 – 255]: 0 = 0% 255 = 100%
0x03 (WO)	CHAR	[dig0] [dig1] [dig2] [dig3]	Dig{0,1,2,3} are ASCII characters
0x04 (WO)	RAW	[dig0] [dig1] [dig2] [dig3]	Dig{0,1,2,3}: bit0 = seg g bit1 = seg f bit2 = seg e bit3 = seg d bit4 = seg c bit5 = seg b bit6 = seg a bit7 = seg dp 
0x05 (WO)	INTEGER	[int MSB] [int LSB]	Signed integer [-999 to 9999]

0x06	BLINK	[period MSB] [period LSB] [duty cycle]	period in ms [0-65535] duty cycle [0-255]: 0 = 0% 255 = 100%

4.5 IRQ

The following table list the commands supported by IRQ register.

Command	**Name**	**Parameters**	**Comment**
0x01 (RO)	REGISTER	[value]	Reset after each reading
0x02	MASK	[mask]	Set the IRQ mask.

Annex A: Robot HMI v1

The Robot HMI I2C address is 0x61.

Pinmux of the Robot HMI

Device	ID	PIN	Comment
LED	0x00	A0	LED yellow
LED	0x01	A1	LED green
LED	0x02	A2	LED blue
LED_RGB	0x00	5 / 6 / 9	LED RGB
BUTTON	0x00	7	Starting cord
BUTTON	0x01	0	Push button 1
BUTTON	0x02	1	Push button 2
BUTTON	0x03	4	Push button 3
BUTTON	0x04	A3	Rocker switch 1
BUTTON	0x05	14	Rocker switch 2
DISPLAY_ 7SEG_4DIG	0x00	10 / 16 /15	Display controlled by MAX7219 connected by SPI

IRQ register (irq pin 8, id 0)

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
		BUTTON 0x05	BUTTON 0x04	BUTTON 0x03	BUTTON 0x02	BUTTON 0x01	BUTTON 0x00