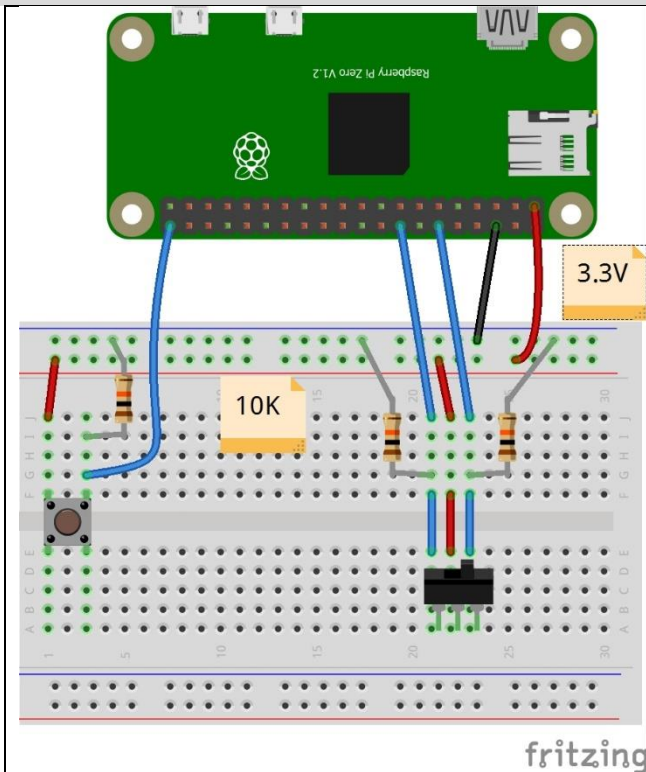


This document lists all the types of Input and Output devices that is handled by the Raspberry Pi gpiozero library. We try to give a simple guide to connect all of the available devices in the gpiozero library.

Button and Digital Input Device



The Button is connected to the ground with a 10K Ohms resistor and to pin 40 of the Raspberry Pi. The other pin of the button is connected to the 3.3V

The Selector is comprised of 2 DigitalInputDevice. The middle pin is connected to the 3.3V and the external pin to the ground with a 10K Ohms resistor. One selector pin is connected to the pin 12 and the other pin 16 of the Raspberry Pi.

You define those like this:

```
<MQTT_Button name="bigButton" pin="40" when_pressed="True" when_held="True" when_released="True">
  <Static pull_up="False" active_state="None" pin_factory="None"/>
  <Time bounce_time="0.05" hold_time="0.25" hold_repeat="False"/>
</MQTT_Button>

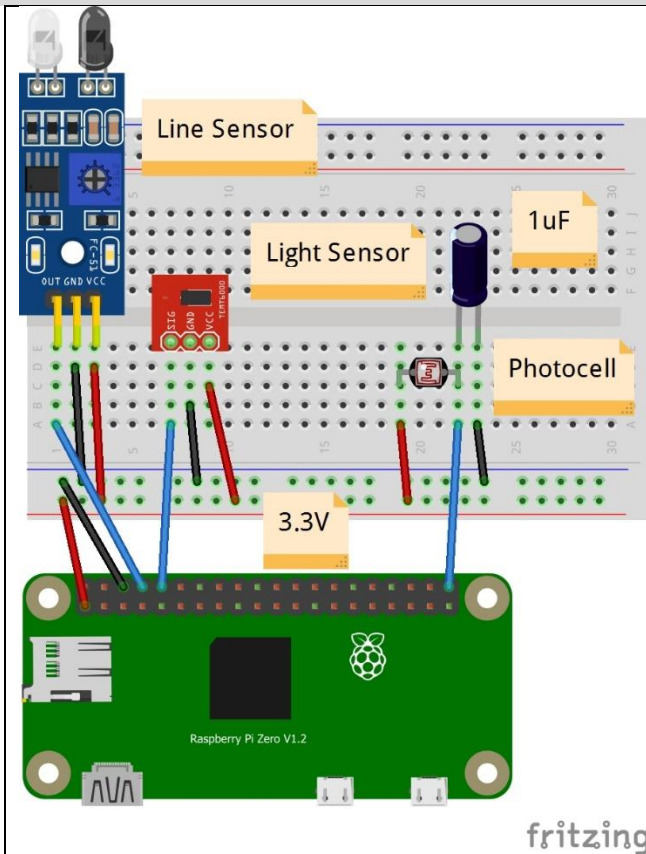
<MQTT_DigitalInputDevice name="selector1" pin="12" when_activated="True" when_deactivated="True" bounce_time="0.02">
  <Static pull_up="True" active_state="None" pin_factory="None"/>
</MQTT_DigitalInputDevice>

<MQTT_DigitalInputDevice name="selector2" pin="16" when_activated="True" when_deactivated="True" bounce_time="0.02">
  <Static pull_up="True" active_state="None" pin_factory="None"/>
</MQTT_DigitalInputDevice>
```

Or more simply, since we use default values:

```
<MQTT_Button name="bigButton" pin="40" when_pressed="True" when_held="True" when_released="True"/>
<MQTT_DigitalInputDevice name="selector1" pin="12" when_activated="True" when_deactivated="True" bounce_time="0.02"/>
<MQTT_DigitalInputDevice name="selector2" pin="16" when_activated="True" when_deactivated="True" bounce_time="0.02"/>
```

Line Sensor and Light Sensor



The line sensor is connected to the 3.3V, the Ground, and to a pin for the signal.

The Light sensor has the same connection profile.

The photocell (LDR) has one pin connected to 3.3V and the other shares the GPIO pin and one of the pin of the 1uF capacitor connected to ground.

You define those like this:

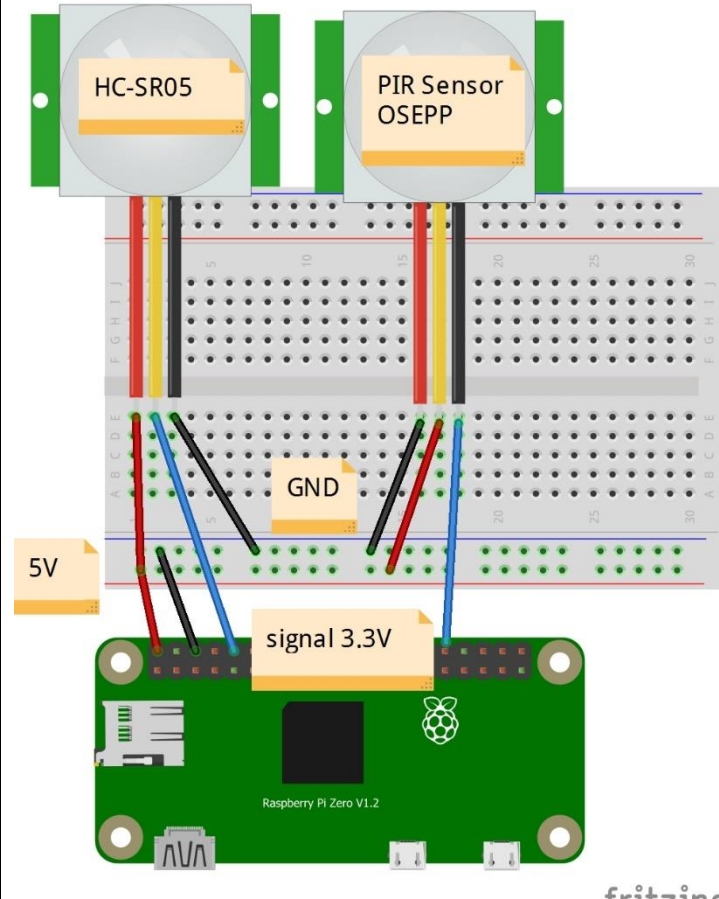
```
<!-- MQTT LineSensor TRCT5000 infra red proximity sensor -->
<MQTT_LineSensor name="LineSensor1" pin="14" when_line="True" when_no_line="True">
  <Static pull_up="False" active_state="None" pin_factory="None"/> <!-- Optional -->
  <Queue queue_len="5" sample_rate="100" threshold="0.5" partial="False"/> <!-- Optional -->
</MQTT_LineSensor>

<!-- MQTT LightSensor -->
<MQTT_LightSensor name="LightSensor1" pin="15" when_light="True" when_dark="True">
  <Static pull_up="False" active_state="None" pin_factory="None"/> <!-- Optional -->
  <Queue queue_len="1" charge_time_limit="0.01" threshold="0.1" partial="False"/> <!-- Optional -->
</MQTT_LightSensor>
<!-- MQTT LightSensor -->
<MQTT_LightSensor name="LS2" pin="16" when_light="True" when_dark="True">
  <Static pull_up="False" active_state="None" pin_factory="None"/> <!-- Optional -->
  <Queue queue_len="1" charge_time_limit="0.01" threshold="0.1" partial="False"/> <!-- Optional -->
</MQTT_LightSensor>
```

Or more simply, since we use default values:

```
<MQTT_LineSensor name="LineSensor1" pin="14" when_line="True" when_no_line="True"/>
<MQTT_LightSensor name="LightSensor1" pin="15" when_light="True" when_dark="True"/>
<MQTT_LightSensor name="LS2" pin="16" when_light="True" when_dark="True"/>
```

Motion Sensor



HC-SR05

PIR Sensor OSEPP


5V

GND


signal 3.3V

Raspberry Pi Zero V1.2

fritzing



<p>Counter-Clockwise or Left</p> <p>Increases Sensitivity. Fully left and the range will be approximately 7 meters.</p>	<p>Clockwise or Right</p> <p>Decreases Sensitivity. Fully right and the range will be approximately 3 meters.</p>
---	---



<p>Counter-Clockwise or Left</p> <p>Decreases Delay Time. Fully left and the delay will be approximately 5 seconds.</p>	<p>Clockwise or Right</p> <p>Increases Delay Time. Fully right and the delay will be approximately 5 minutes.</p>
---	---

Both types of PIR use 5V as input but gives out 3.3V signals. No resistor needed.

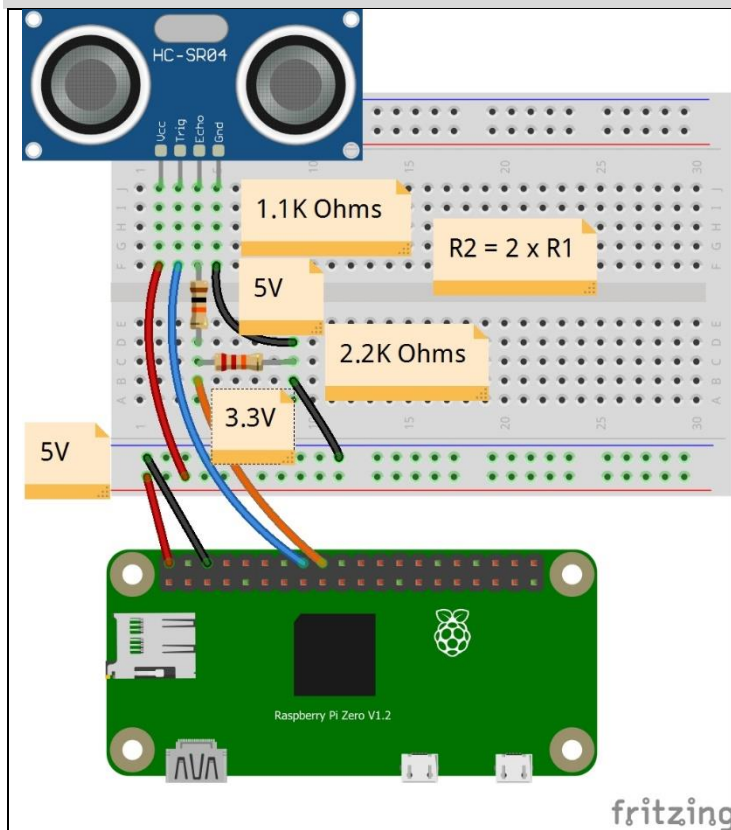
You define those like this:

```
<!-- MQTT MotionSensor D-SUN PIR (Passive IR) motion sensor -->
<MQTT_MotionSensor name="PIR1" pin="12" when_motion="True" when_no_motion="True">
  <Static pull_up="False" active_state="None" pin_factory="None"/> <!-- Optional -->
  <Queue queue_len="1" sample_rate="10" threshold="0.5" partial="False"/> <!-- Optional -->
</MQTT_MotionSensor>
<MQTT_MotionSensor name="PIR2" pin="14" when_motion="True" when_no_motion="True">
  <Static pull_up="False" active_state="None" pin_factory="None"/> <!-- Optional -->
  <Queue queue_len="1" sample_rate="10" threshold="0.5" partial="False"/> <!-- Optional -->
</MQTT_MotionSensor>
```

Or more simply, since we use default values:

```
<MQTT_MotionSensor name="PIR1" pin="12" when_motion="True" when_no_motion="True"/>
<MQTT_MotionSensor name="PIR2" pin="14" when_motion="True" when_no_motion="True"/>
```

Distance Sensor



The distance sensor has two pins, trigger (output) and echo (input). It uses 5V power and has a 5V output signal that need a voltage divider where $R2 = 2 \times R1$ to have a 3.3V to the GPIO pin.

The loop_process sends the distance information at every X seconds, defined in the XML.

You define those like this:

```
<!-- MQTT DistanceSensor HC-SR04 -->
<MQTT_DistanceSensor name="DistanceSensor1" echo="12" trigger="6" when_in_range="True" when_out_of_range="True" send_every="None">
  <Static pin_factory="None"/> <!-- Optional -->
  <Queue queue_len="30" max_distance="1" threshold_distance="0.3" partial="False" /> <!-- Optional -->
</MQTT_DistanceSensor>
```

Or more simply, since we use default values:

```
<MQTT_DistanceSensor name="DistanceSensor1" echo="12" trigger="6" when_in_range="True" when_out_of_range="True" send_every="None"/>
```

Keypad

--	--

You define those like this:

Or more simply, since we use default values:

RFID

--	--

You define those like this:

Or more simply, since we use default values:

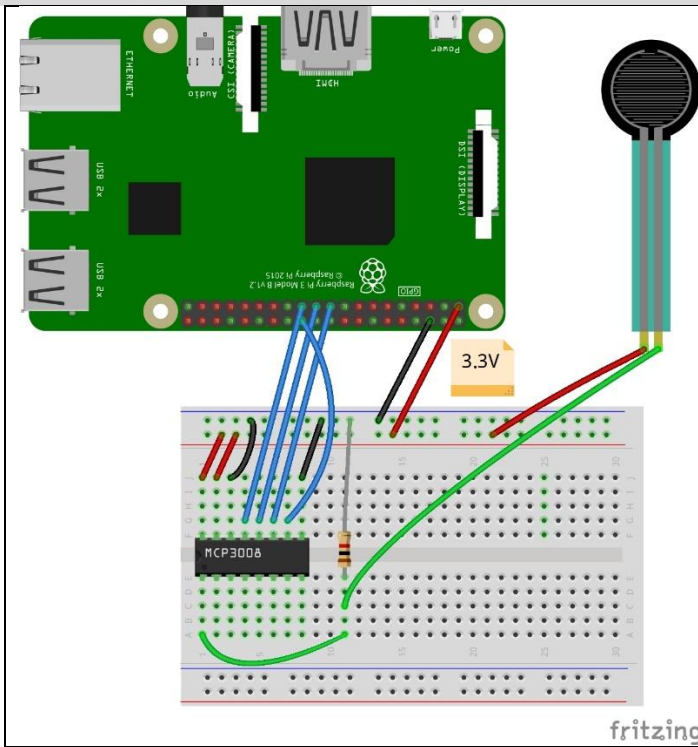
USB-GPS

--	--

You define those like this:

Or more simply, since we use default values:

MCP3008



You define those like this:

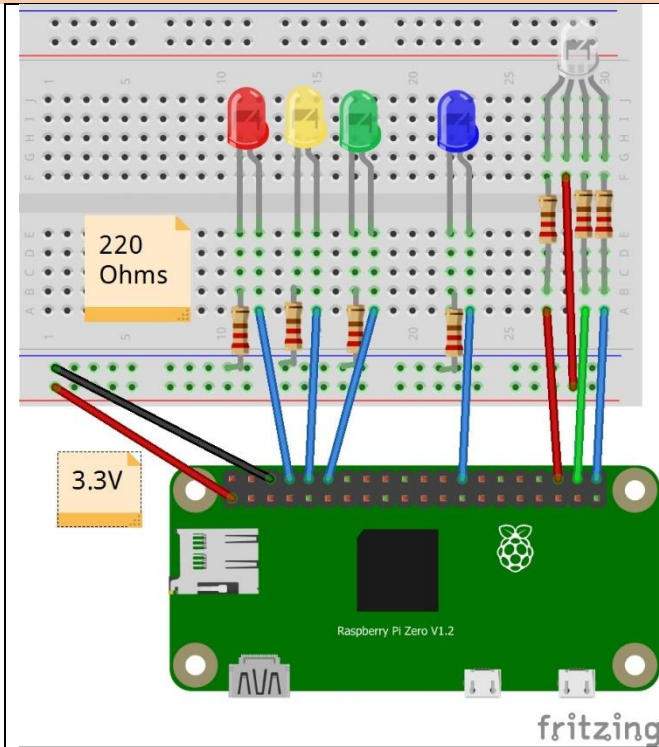
Or more simply, since we use default values:

Pi Camera

You define those like this:

Or more simply, since we use default values:

Led, PWMLed, RGBLed



All connections to ground is using a 220 Ohms resistor, even those for the RGB Led.

There are 3 LEDs, one PWM LED, and one RGB LED.

All are using 3.3V

You define those like this:

```
<MQTT_LED name="Red1" pin="8">
  <Static active_high="True" initial_value="False" pin_factory="None"/>
</MQTT_LED>
<MQTT_LED name="Yellow1" pin="10">
  <Static active_high="True" initial_value="False" pin_factory="None"/>
</MQTT_LED>
<MQTT_LED name="Green1" pin="12">
  <Static active_high="True" initial_value="False" pin_factory="None"/>
</MQTT_LED>

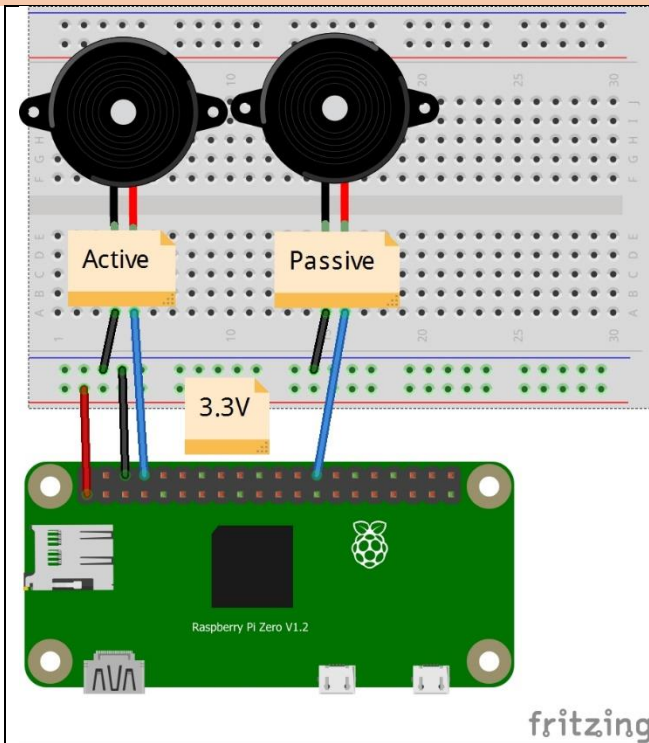
<MQTT_PWMLED name="PWM_Blue1" pin="26">
  <Static active_high="True" initial_value="0" frequency="100" pin_factory="None"/>
</MQTT_PWMLED>

<MQTT_RGBLED name="RGB1" r="36" g="38" b="40">
  <Static active_high="True" initial_value="(0, 0, 0)" pwm="True" pin_factory="None"/>
</MQTT_RGBLED>
```

Or more simply, since we use default values:

```
<MQTT_LED name="Red1" pin="8"/>
<MQTT_LED name="Yellow1" pin="10"/>
<MQTT_LED name="Green1" pin="12"/>
<MQTT_PWMLED name="PWM_Blue1" pin="26"/>
<MQTT_RGBLED name="RGB1" r="36" g="38" b="40"/>
```


Buzzer, Tonal Buzzer



Be it Active or Passive Buzzer, it works with 3.3V.

Connecting it is very simple. Both Buzzer and Tonal Buzzer connect the same. One pin to ground, the other to a GPIO.

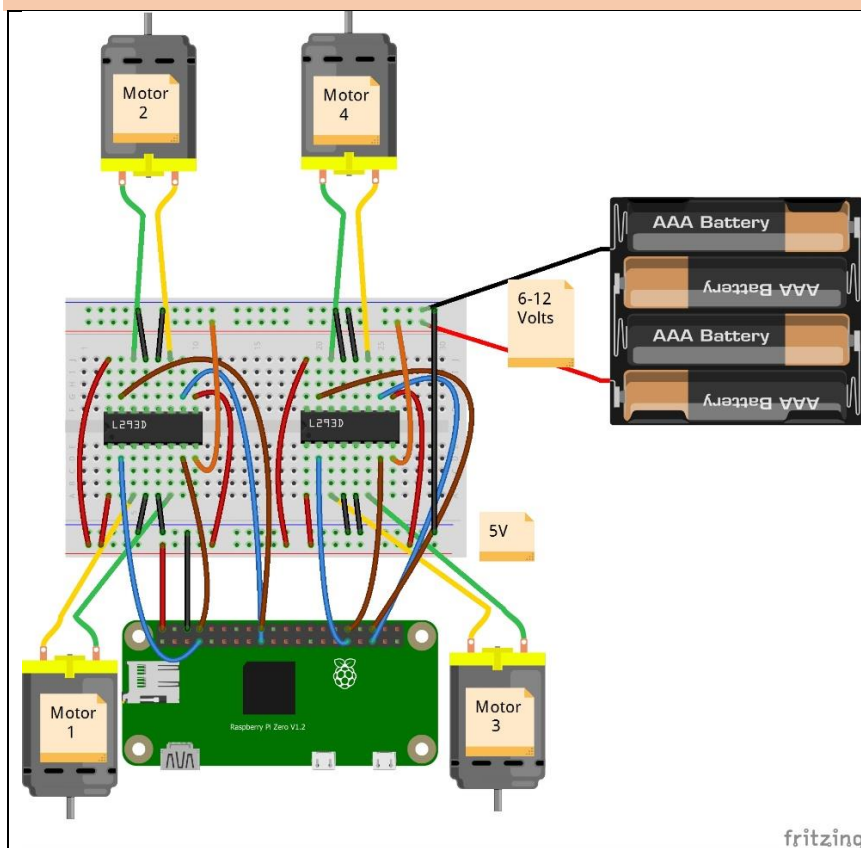
You define those like this:

```
<!-- MQTT Buzzer -->
<MQTT_Buzzer name="Buzzer1" pin="12">
  <Static active_high="True" initial_value="False" pin_factory="None"/> <!-- Optional -->
</MQTT_Buzzer>
<!-- MQTT TonalBuzzer -->
<MQTT_TonalBuzzer name="TonalBuzzer1" pin="14">
  <ToneSpec initial_value="None" mid_tone="A4" octaves="1"/> <!-- Optional -->
  <Static active_high="True" pin_factory="None"/> <!-- Optional -->
</MQTT_TonalBuzzer>
```

Or more simply, since we use default values:

```
<MQTT_Buzzer name="Buzzer1" pin="12"/>
<MQTT_TonalBuzzer name="ToanlBuzzer1" pin="12"/>
```


Motor



Connecting 4 motors using 2 L293D chips.

Motor 1&2 or Motor 3&4. Each L293D is connected as followed, one for each pair of motor :

Pin 1 : enable 1,2	Pin9: enable 3,4
Pin2: input 1	Pin10: input 3
Pin3 Output 1	Pin11 Output 3
Pin4-5: GND	Pin12-13: GND
Pin6: Output 2	Pin14: Output 4
Pin7: Input 2	Pin15: Input 4
Pin8: VCC2	Pin16: VCC1

VCC1 = VCC2 = 6 or 12 volts (depending on the motor needs)

Outputs are connected directly to the motors, Inputs are connected to GPIOs.

The logical enabling pins use 5V to activate.

You define those like this:

```
<!-- MQTT Motor -->
<MQTT_Motor name="Motor1" forward="12" backward="6" enable="None" pwm="True">
  <Static active_high="True" pin_factory="None"/> <!-- Optional -->
</MQTT_Motor>
```

Or more simply, since we use default values:

```
<MQTT_Motor name="Motor1" forward="12" backward="6" enable="None" pwm="True"/>
```

Phase Enable Motor

--	--

You define those like this:

```
<!-- MQTT PhaseEnableMotor -->
<MQTT_PhaseEnableMotor name="PhaseEnableMotor1" phase="12" enable="13" pwm="True">
  <Static active_high="True" pin_factory="None"/> <!-- Optional -->
</MQTT_PhaseEnableMotor>
```

Or more simply, since we use default values:

```
<MQTT_PhaseEnableMotor name="PhaseEnableMotor1" phase="12" enable="13" pwm="True"/>
```

Servo

--	--

You define those like this:

```
<!-- MQTT Servo -->
<MQTT_Servo name="Servo1" pin="12" initial_value="0" min_pulse_width="1/1000" max_pulse_width="2/1000" frame_width="20/1000">
  <Static active_high="True" pin_factory="None"/> <!-- Optional -->
</MQTT_Servo>
```

Or more simply, since we use default values:

```
<MQTT_Servo name="Servo1" pin="12" initial_value="0" min_pulse_width="1/1000" max_pulse_width="2/1000" frame_width="20/1000"/>
```

Angular Servo

--	--

You define those like this:

```
<!-- MQTT AngularServo -->
<MQTT_AngularServo name="AngularServo1" pin="12" initial_angle="0" min_angle="-90" max_angle="90" min_pulse_width="1/1000"
  max_pulse_width="2/1000" frame_width="20/1000">
  <Static active_high="True" pin_factory="None"/> <!-- Optional -->
</MQTT_AngularServo>
```

Or more simply, since we use default values:

```
<MQTT_AngularServo name="AngularServo1" pin="12" initial_angle="0" min_angle="-90" max_angle="90" min_pulse_width="1/1000"
  max_pulse_width="2/1000" frame_width="20/1000"/>
```

Digital Output Device

--	--

You define those like this:

```
<!-- MQTT DigitalOutputDevice -->
<MQTT_DigitalOutputDevice name="DigitalOutputDevice1" pin="12">
  <Static active_high="True" initial_value="False" pin_factory="None"/> <!-- Optional -->
</MQTT_DigitalOutputDevice>
```

Or more simply, since we use default values:

```
<MQTT_DigitalOutputDevice name="DigitalOutputDevice1" pin="12"/>
```

PWM Output Device

--	--

You define those like this:

```
<!-- MQTT PWMOutputDevice -->
<MQTT_PWMOutputDevice name="PWMOutDev1" pin="12">
  <Static active_high="True" initial_value="0" frequency="100" pin_factory="None"/> <!-- Optional -->
</MQTT_PWMOutputDevice>
```

Or more simply, since we use default values:

```
<MQTT_PWMOutputDevice name="PWMOutDev1" pin="12"/>
```

Output Device

--	--

You define those like this:

```
<!-- MQTT OutputDevice -->
<MQTT_OutputDevice name="OutputDevice1" pin="12">
  <Static active_high="True" initial_value="False" pin_factory="None"/> <!-- Optional -->
</MQTT_OutputDevice>
```

Or more simply, since we use default values:

```
<MQTT_OutputDevice name="OutputDevice1" pin="12"/>
```

Sound Mixer

--	--

You define those like this:

```
<!-- MQTT SoundMixer -->  
<MQTT_SoundMixer name="SoundMixer1">  
  <Static nbr_of_channels="6" frequency="48000" size="-16" mono_or_stereo="1" buffer_size="1024"/> <!-- Optional -->  
</MQTT_SoundMixer>
```

Or more simply, since we use default values:

```
<MQTT_SoundMixer name="SoundMixer1"/>
```

2x16 LCD I2C

--	--

You define those like this:

```
<!-- MQTT 2x16_lcd_I2C (uses pins: SDA1(3), SCL1(5). GND, 5V) -->  
<MQTT_2x16_lcd_I2C name="LCD1" address="0x27"/>
```

Shift Register

--	--

You define those like this:

Or more simply, since we use default values: