IBIS

User’s Manual

Version A – 10/10/2021 (DRW)

System Configuration

For the system to know what is being controlled, the system will need to be defined along with each device and how it’s connected to the mother controller as well as any children controllers and the devices that each must manage.

System Configuration File

The system configuration file is an XML file located as ‘***/etc/IBIS/conf/system.xml’***, any child controller will receive its instructions from this file as well through the initialization of that device.

Definition

XML Header

It is not the goal of this manual to explain what an XML file is and how to format it. But if you follow the formats specified in this manual you probably don’t even need to know the details.

The first line should have the following stated in this text file, it basically states that it’s an XML file version 1.0.

<?xml version="1.0"?>

System Tag

This defines the name of the system and the identifying number that will be used when each system communicates with each other.

The ***name*** attribute is only used for displaying and logging purposes and should be short and descriptive.

The ***id*** attribute should be unique for each controller as it's what will be used to allow each system to identify each other with.

```
<System name="Main Control" id="100">

</System>
```

Devices Tag

This is an additional entry inside of the System tag defined above you will need to insert a line before the </System> tag line is specified.

```
<System name="Main Control" id="100">

  <Devices>

  </Devices>

</System>
```

Device Tag

Each device defined will have an entry in between the <Devices> and </Devices> tags.

```
<System name="Main Control" id="100">

  <Devices>

    <Device id="1" name="BLUE " type="LED" PIN="0" INITSTATE="LOW" INITMODE="OUTPUT" STATES="Blink1" />

  </Devices>

</System>
```

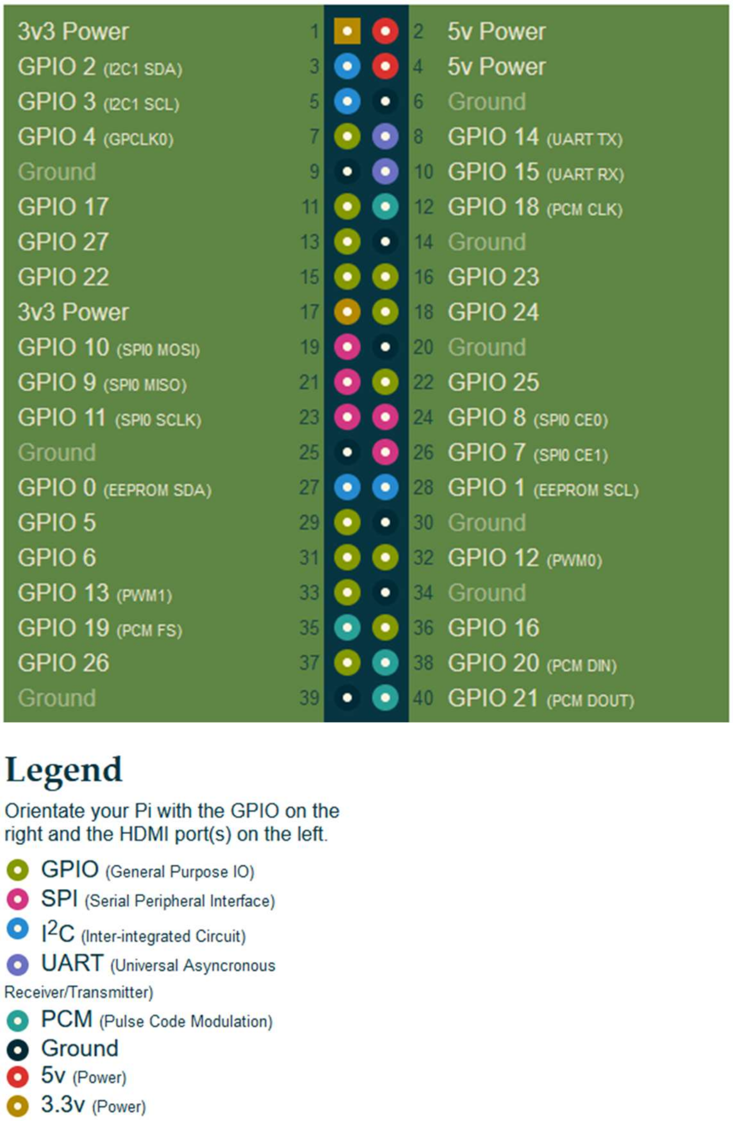
The ***id*** attribute is used to uniquely define each device within the system, each device should have a unique id to allow you to create triggers and other dependencies that will be defined in later versions of this program.

The ***name*** attribute is used primarily for display and logging purposes only, a good example is to use the color of a LED light, or which side of the car the motor is on, etc.

The ***type*** attribute allows the system to know which type of device this is, please note that this attribute will need to match the behavior "state machine" programming configuration. Here is the current table that can be used:

Device Type	Description
LED	This is a normal LED light

The PIN attribute is used to let the system know which GPIO pin is used for this device, currently the system is based on the "Wired PI" libraries and the best place to get this information dynamically as you are looking at the board would be at this URL https://pinout.xyz/# because you can get the image shown below and you can click on any pin to find out the translation.



3v3 Power	1	2	5v Power
GPIO 2 (I2C1 SDA)	3	4	5v Power
GPIO 3 (I2C1 SCL)	5	6	Ground
GPIO 4 (GPCLK0)	7	8	GPIO 14 (UART TX)
Ground	9	10	GPIO 15 (UART RX)
GPIO 17	11	12	GPIO 18 (PCM CLK)
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3v3 Power	17	18	GPIO 24
GPIO 10 (SPI0 MOSI)	19	20	Ground
GPIO 9 (SPI0 MISO)	21	22	GPIO 25
GPIO 11 (SPI0 SCLK)	23	24	GPIO 8 (SPI0 CE0)
Ground	25	26	GPIO 7 (SPI0 CE1)
GPIO 0 (EEPROM SDA)	27	28	GPIO 1 (EEPROM SCL)
GPIO 5	29	30	Ground
GPIO 6	31	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	34	Ground
GPIO 19 (PCM FS)	35	36	GPIO 16
GPIO 26	37	38	GPIO 20 (PCM DIN)
Ground	39	40	GPIO 21 (PCM DOUT)

Legend

Orientate your Pi with the GPIO on the right and the HDMI port(s) on the left.

- ● GPIO (General Purpose IO)
- ● SPI (Serial Peripheral Interface)
- ● I^{2}C (Inter-integrated Circuit)
- ● UART (Universal Asyncronous Receiver/Transmitter)
- ● PCM (Pulse Code Modulation)
- ● Ground
- ● 5v (Power)
- ● 3.3v (Power)

Wiring Pi pin	GPIO	Physical Board
0	17	11
1	18	12
2	27	13
3	22	15
4	23	16
5	24	18
6	25	22
7	4	7
8	2	3

9	3	5
10	8	24
11	7	26
12	10	19
13	9	21
14	11	23
15	14	8
16	15	10
21	5	29
22	6	31
23	13	33
24	19	35
25	26	37
26	12	32
27	16	36
28	20	38
29	21	40
30	0	27
31	1	28

The attribute ***INITSTATE*** tells the system what state is desired for this device when it is first initialized, this is only valid if the ***INITMODE*** is equal to ***OUTPUT*** instead of ***INPUT***.

The attribute ***INITMODE*** defines what mode this device should be when it's initialized.

The attribute ***STATES*** is used to specify what behavior-based state machine program will be used for this device.

State Machine Behavior Program

Each device type can have as many state machines defined for use by any device of the type specified. At anytime either the system or another device can reassign the behavior of a device, this allowed for things like having an LED off until someone pushes a button, or stopping a motor because it's about to hit a wall, etc. While these concepts are not currently defined in this version of the system, Links and Triggers will be added later.

Program Configuration File

The state machine program configuration files are XML files located at '***/etc/IBIS/conf.d/'***, any file name can be used if it ends with .xml. Typically, the file would match the name of the program, such as "/etc/IBIS/conf.d/FastBlink1.xml" would be used if the program name was called "FastBlink1" that way it's easy to figure out which file is used for which program.

Definition

XML Header

It is not the goal of this manual to explain what an XML file is and how to format it. But if you follow the formats specified in this manual you probably don't even need to know the details.

The first line should have the following stated in this text file, it basically states that it's an XML file version 1.0.

```
<?xml version="1.0"?>
```

Program Tag

This defines the name of the program and the device type that this program was designed for. Please note that a program designed for an LED light should not be used for a device of a different type as the behavior will not match what you would expect.

The ***name*** attribute is used explicitly to define the behavior of a device and is used every time this is changed.

The ***type*** attribute should match the device type that this program was designed for, if you want to define what a LED light does, then this would be LED. See the System LED attribute to see all options for this version of the program.

States Tag

This is an additional entry inside of the Program tag defined above you will need to insert a line before the </Program> tag line is specified.

```
<Program name="FastBlink1" type="LED">
  <States>
  </States>
</Program>
```

State Tag

Each state entry will define the behavior of the device when assigned this program. Each entry will be inserted between the <States> and </States> tags.

```
<Program name="FastBlink1" type="LED">
    <States>
```

```
        <State id="1" nextID="2" name="LED On" code="@" delaySeconds="0" delayMS="100" PINState="1" />

        <State id="2" nextID="1" name="LED Off" code="@" delaySeconds="0" delayMS="100" PINState="0" />

    </States>

</Program>
```

The ***id*** attribute is a line number or state number allowing the ability to go from one state to another using this id to be referenced by.

The ***nextID*** attribute is used to specify what the next id will be when this completed based on the condition specified in the code attribute.

The ***name*** attribute is only used for display and logging purposes.

The ***code*** attribute is used to define the condition used to determine the next state.

Code Value	Definition
@	This is an unconditional code, telling the system that when this state is completed immediately go to the next id specified in the ***nextID*** attribute.

The ***delaySeconds*** attribute specifies how long in seconds that this state will be in its current state if uninterrupted.

The ***delayMS*** attribute can be used to add milliseconds to the ***delaySeconds*** attribute for higher resolution delays.

The ***PINState*** attribute is used to change the state of the device at the beginning of the state.