These slides are used in the Open Mission Stack Raspberry Pi App Demo project at

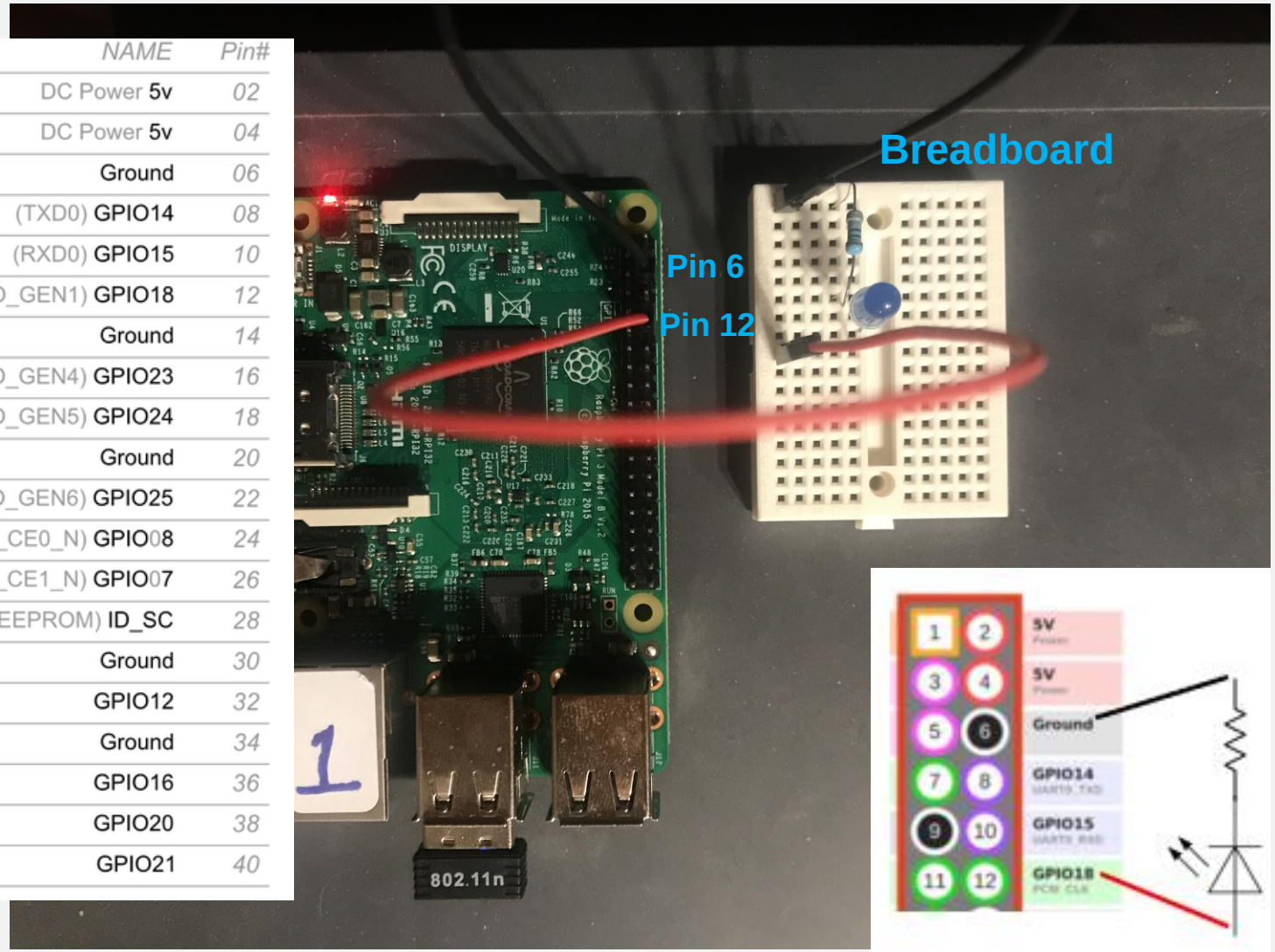
https://openmissionstack.com/projects_read/gpio_demo

They are included in GPIO_DEMO’s repo as a reference. See the project website for detailed instructions.

Configure and connect a breadboard to the Pi as shown below

- Note physical pin 12 is logical GPIO pin 18 which is the identifier used by the FSW

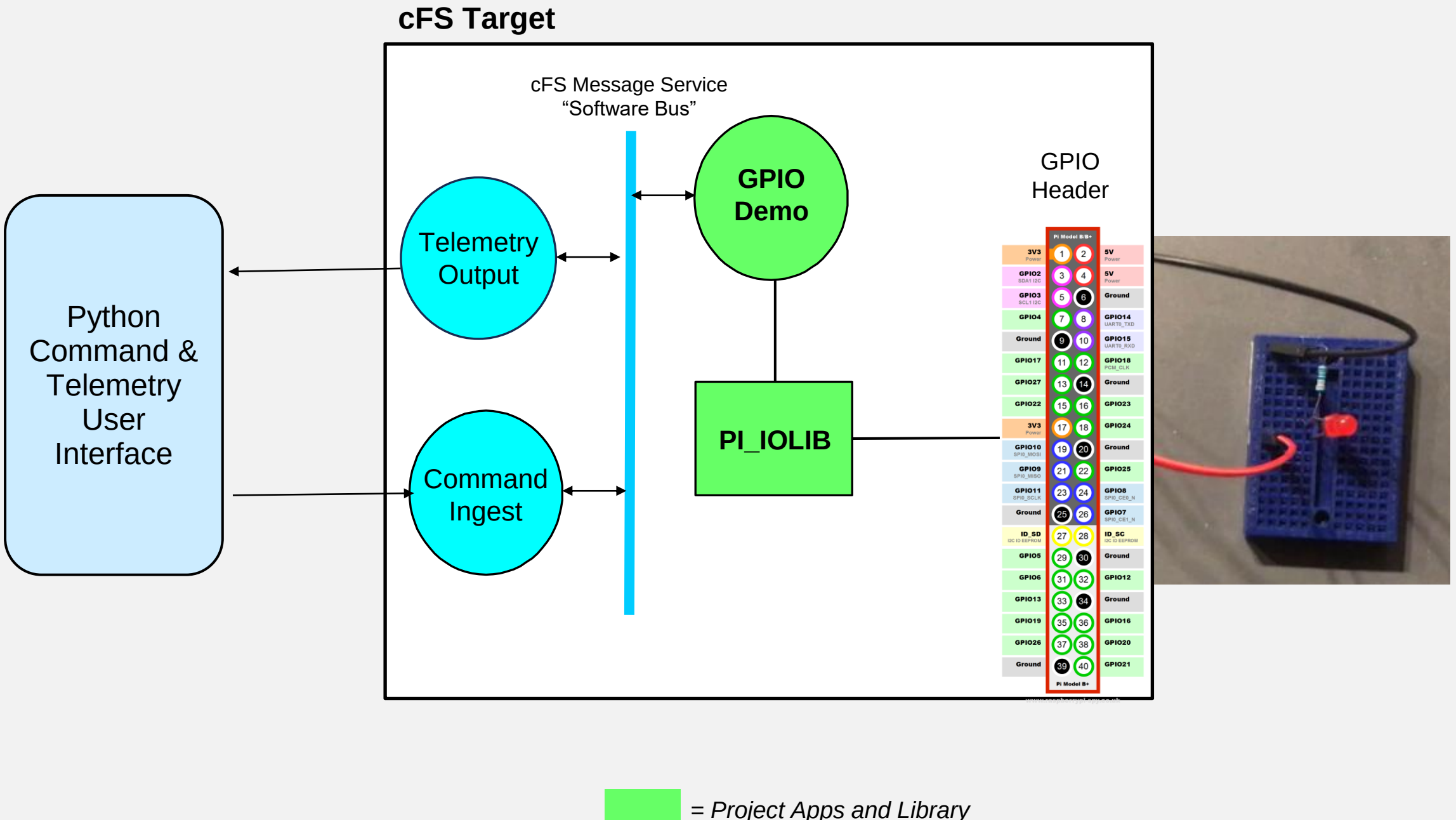
Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I2C)		DC Power 5v	04
05	GPIO03 (SCL1 , I2C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)		(I2C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40



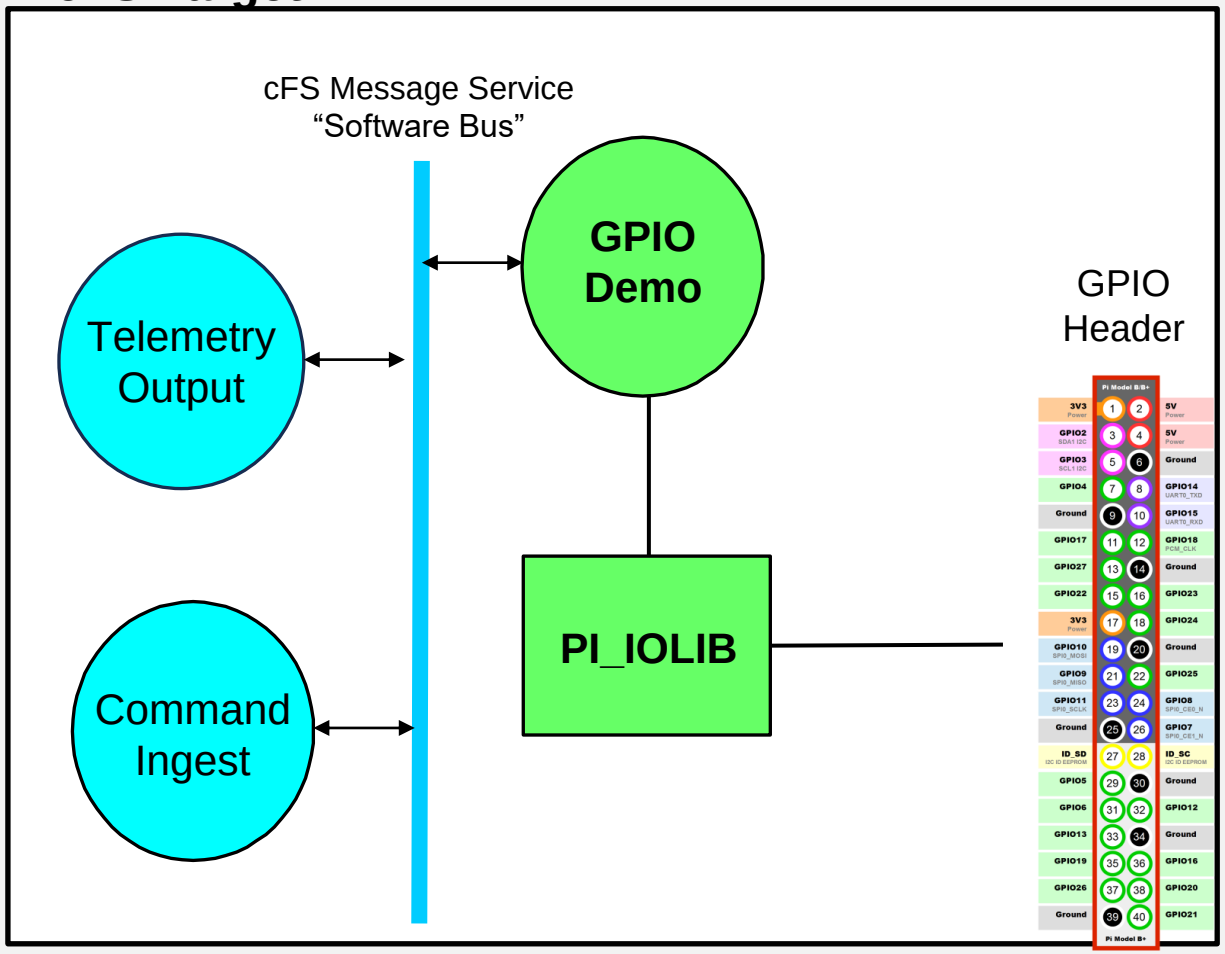
220 Ohm Resister
(red, red, brown)

LED
- *Connect shorter leg Cathode (-) to resistor*
- *Connect longer leg Anode (+) to GPIO pin 12*

Connect LED to RPI GPIO header


cFS Target
cFS Message Service
“Software Bus”
Python Command & Telemetry User Interface
Telemetry Output
Command Ingest
GPIO Demo
PI_IOLIB
GPIO Header
= Project Apps and Library

cFS Target


cFS Message Service
“Software Bus”
GPIO
Demo
Telemetry
Output
Command
Ingest
PI_IOLIB
GPIO
Header


GitHub


Download App
Select one or more apps to download then follow the steps in 'Add App'. See 'A
BERRY_IMU Core Flight System app that interfaces to the Ozzmaker Berry
GPIO_DEMO Example app controlling a GPIO pin that can be used as a start
MQTT_GW Provide a gateway between the core Flight System (cFS) Softw
MQTT_LIB The Eclipse Paho Embedded C MQTT Library ported to the cor
PI_IOLIB cFS library providing an interface to Raspberry Pi peripherals
PL_MGR Example payload management app.
PL_SIM Payload simulator app that provides a ground interface to the
PL_SIM_LIB Payload simulator library that simulates a fictitious detector th
SC_SIM Simulate a simple spacecraft operational interface that expose
TBL_SAT Raspberry Pi app used in the Table Sat kit.
Download
Cancel

GPIO_DEMO App

Commands

- Set LED On Time
- Set LED Off Time

Telemetry

```
HkTlm.Payload.ValidCmdCnt  : 0
HkTlm.Payload.InvalidCmdCnt: 0
HkTlm.Payload.CtrlIsMapped : FALSE
HkTlm.Payload.CtrlOutPin   : 18
HkTlm.Payload.CtrlLedOn    : FALSE
HkTlm.Payload.CtrlSpare    : 0
HkTlm.Payload.CtrlOnTime   : 3000
HkTlm.Payload.CtrlOffTime  : 6000
```

Library and Application Summary

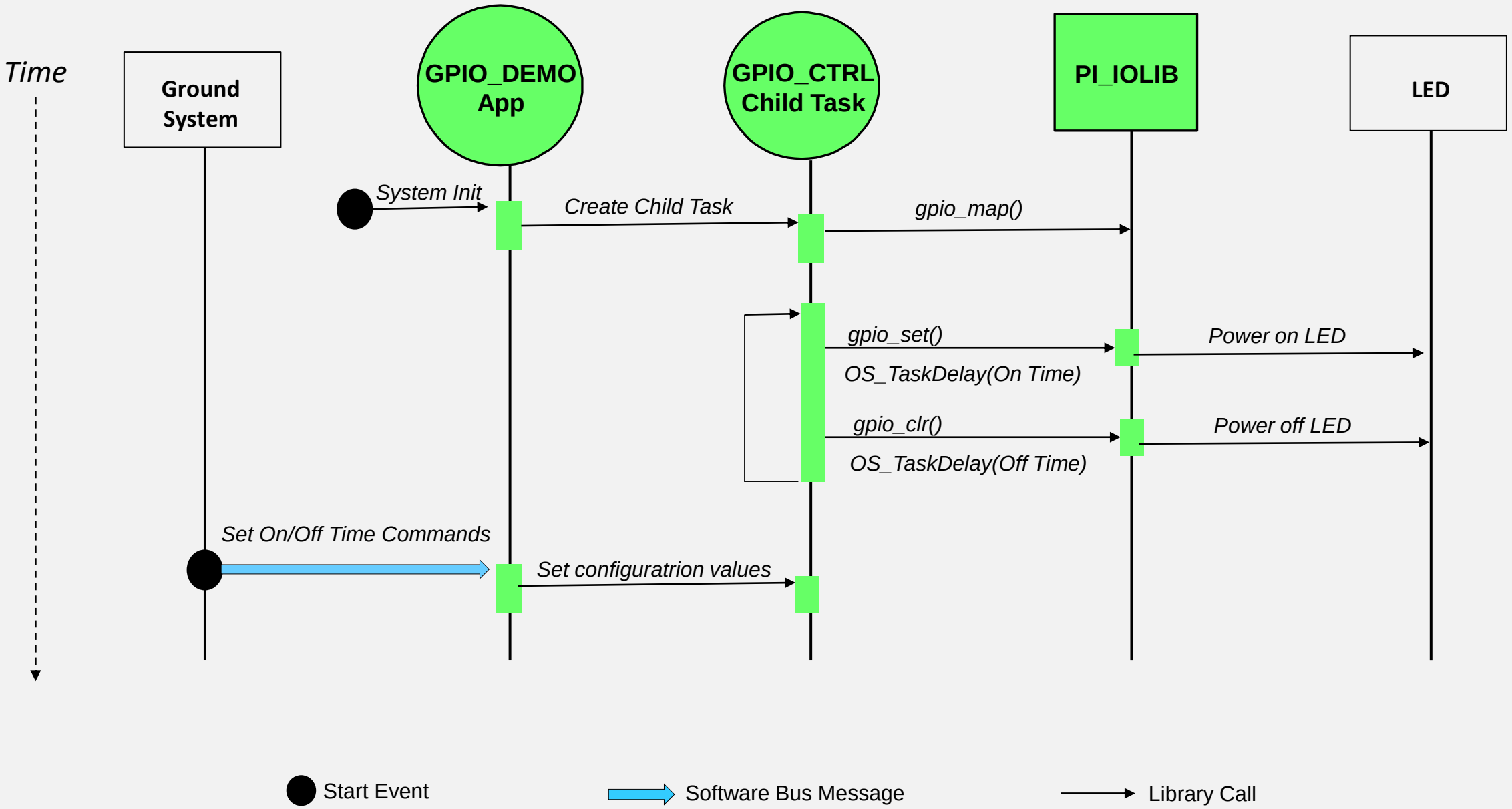
PI_IOLIB

- This library adapts the “minimalistic peripheral access” (MIPEA) library for the Raspberry Pi, https://github.com/jasLogic/mipea so it can serve as a cFS library.
- It provides an interface to the Rasperry Pi’s memory mapped General Purpose I/O registers
- The config.h configuration file allows you to select your Broadcom processor chip

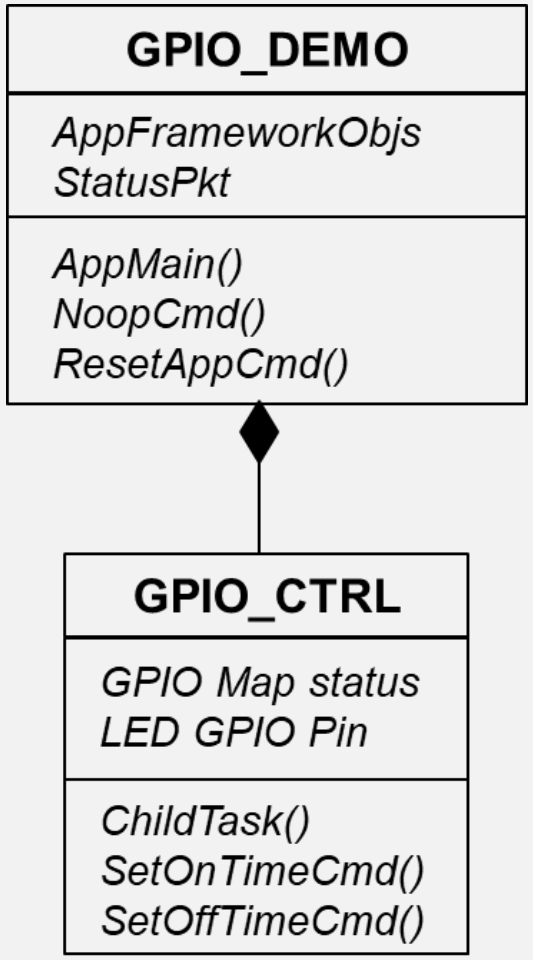
GPIO_DEMO

- The main app manages the ground command and telemetry interface
- A child task controls turning on and off the LED
- Ground commands can be sent to set the LED on and off time durations
- JSON initialization table defines the GPIO LED control pin and the default LED on/off time durations

Control Flows



GPIO_DEMO App Object Design



GPIO_DEMO

- Manages app initialization, main runtime loop, and status telemetry
- Dispatches commands to objects

GPIO_CTRL

- Manages GPIO and LED interface
- Contains child task that turns the LED on and off
- Calls to PI_IOLIB functions should be limited to this object to localize coupling

GPIO Demo Control Flow