

RPi Automation Manual

Draft 1 (Aug 2021)



Table of Contents

Hardware.....	2
UB Board.....	2
REB I/O Board.....	3
AC Solid state relay.....	4
DC solid state relay.....	4
I2C I/O Board.....	5
Sensors.....	5
BME280.....	5
SH1106 OLED.....	6
To do list.....	6
Directory structure.....	7
User data file (user.dat).....	7
Global variables.....	8
RPi specific variables.....	8
Task variables.....	8
GPIO task variables.....	8
List of available functions.....	8
BME280.....	8
I/O read.....	9
Latch.....	9
Manage Disk Space.....	9
Record Camera.....	9
Samba Client.....	9
SH1106 OLED Display.....	9
Thermostat.....	10
Update files.....	10
Watchdog.....	10
Startup.....	10
Loop.....	10
Functions.....	11
RPi Interface.....	11
Startup.....	11
Loop.....	11
Functions.....	11
manage_disk_space(var_data).....	11
record_cam(var_data).....	12
samba_client(var_data).....	13
samba_server(var_data).....	13
test_connectivity(var_data).....	14
update_files(var_data).....	14
wu_forecast(var_data).....	15
wu_upload(var_data).....	15

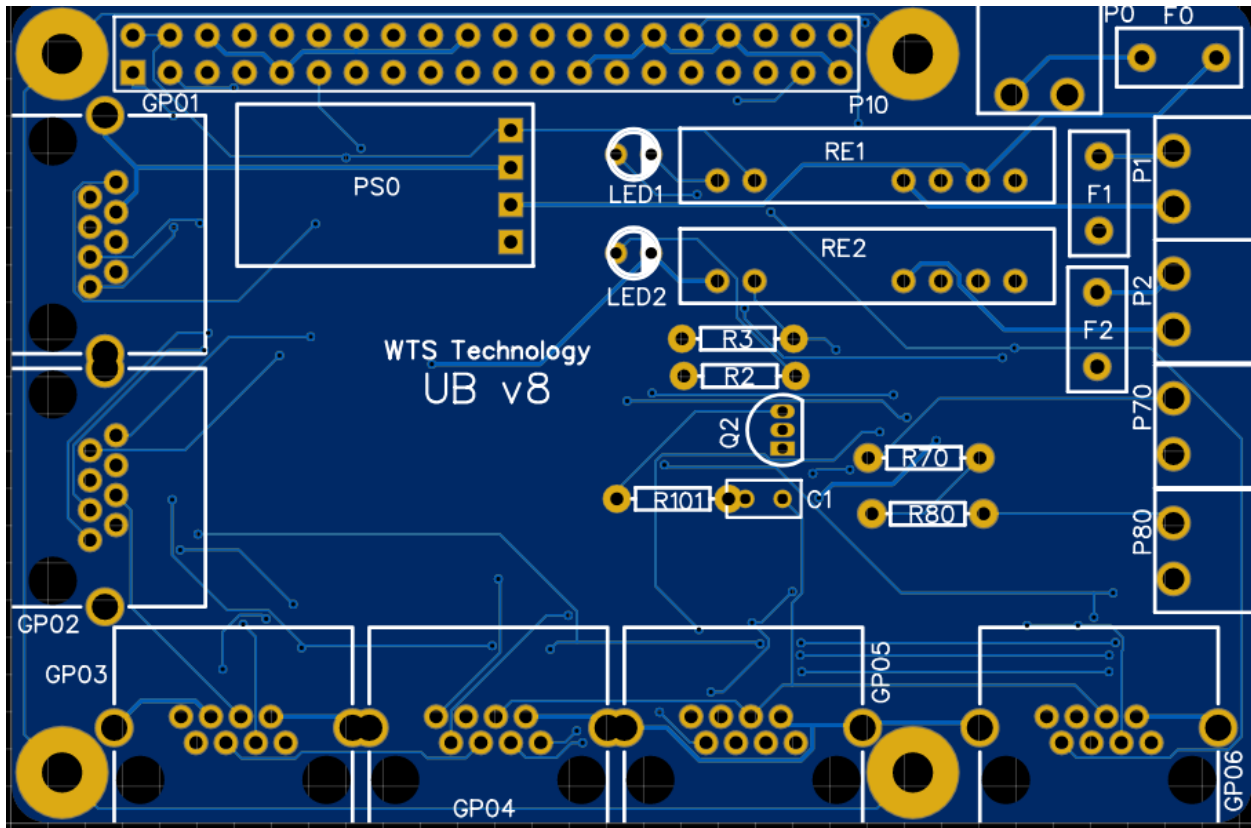


URB Interface.....	17
Startup.....	17
Loop.....	17
Functions.....	18
i2c_accessory(var_data).....	18
i2c_bme280(var_data).....	19
i2c_detect().....	19
i2c_io_read(var_data).....	19
i2c_latch(var_data).....	20
i2c_pcf8574(enable_pin,address,bit,bit_x).....	21
i2c_pcf8574_read(enable_pin,address,bit).....	21
i2c_pins_disable().....	22
i2c_sh1106(var_data).....	23
i2c_timer(var_data).....	24
i2c_thermostat(var_data).....	25
Utility file (utils.py).....	26
Functions.....	26
bash_command(bash_input).....	26
dumb_code(a, b).....	26
file_exists(path_file).....	26
file_get_mod_time(path_file).....	26
file_read(path_file).....	26
file_write(path_file).....	26
i2c_device().....	26
log(data_text, event_file, log_mode).....	26
lookup_text(x_data, text).....	27
lookup_var(x_data, x_search).....	27
pid_util(pid_name, selection).....	27
send_zoho_email(subject, text, event_file, error_file, log_mode).....	27
signed_short(msb, lsb).....	27
write_data(data):.....	27
Old file.....	28
i2c_tsl2561.....	29
Functions.....	29

Hardware

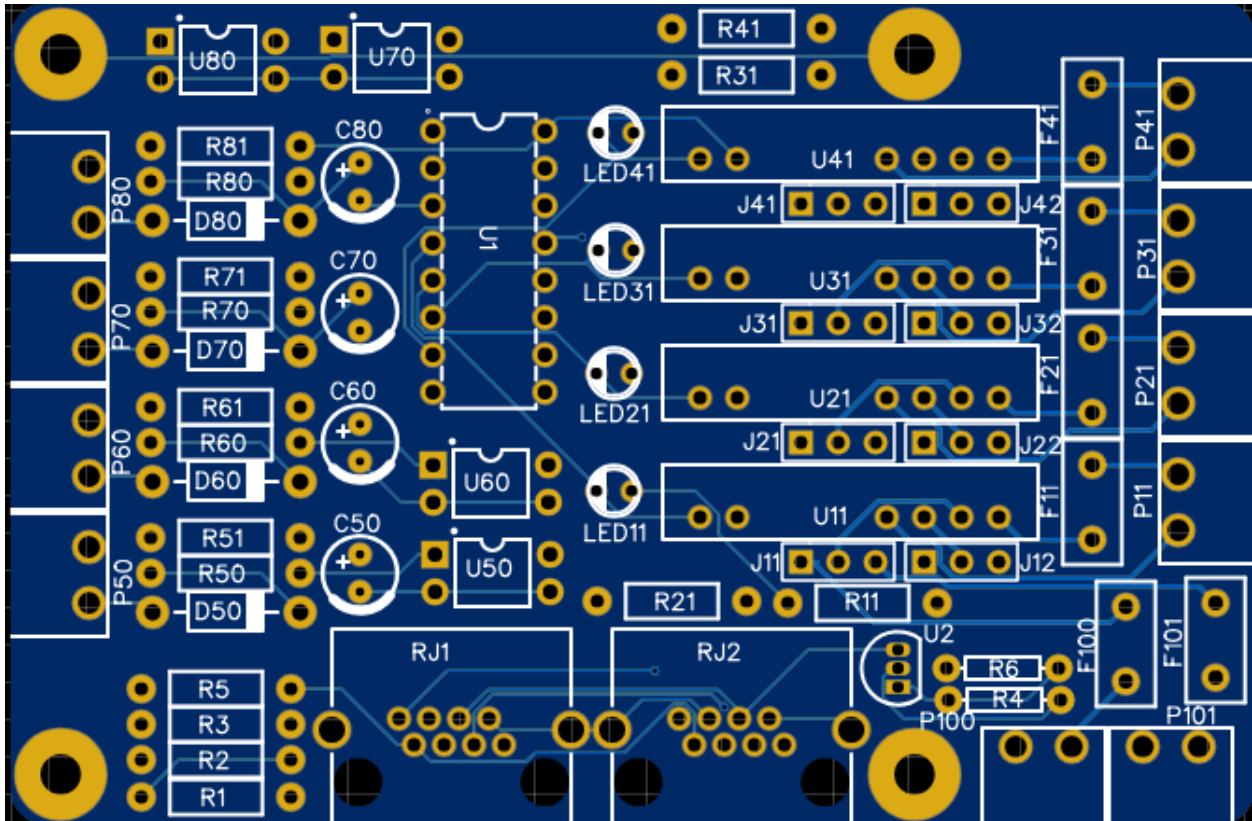
UB Board

- Requires a 12 VDC at 2A power source (P0)
- Generates the 5 VDC for the RPi (PS0)
- The UB board has 6 independent GPIO port (GP01 to GP06).
- Each GP port includes selectable I2C bus and 2 GPIO input / output
- GP6 I2C can be used from the UB board to control 2 relay (RE1, RE2) and 2 contact input.



REB I/O Board

- The board has 4 configurable input for contact switch, AC or DC level and 4 relay output



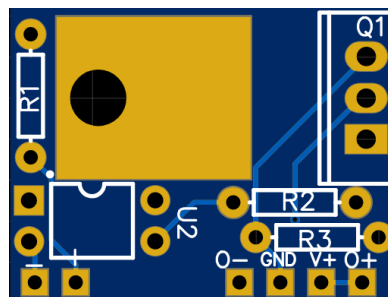
Jumper setting

AC Solid state relay



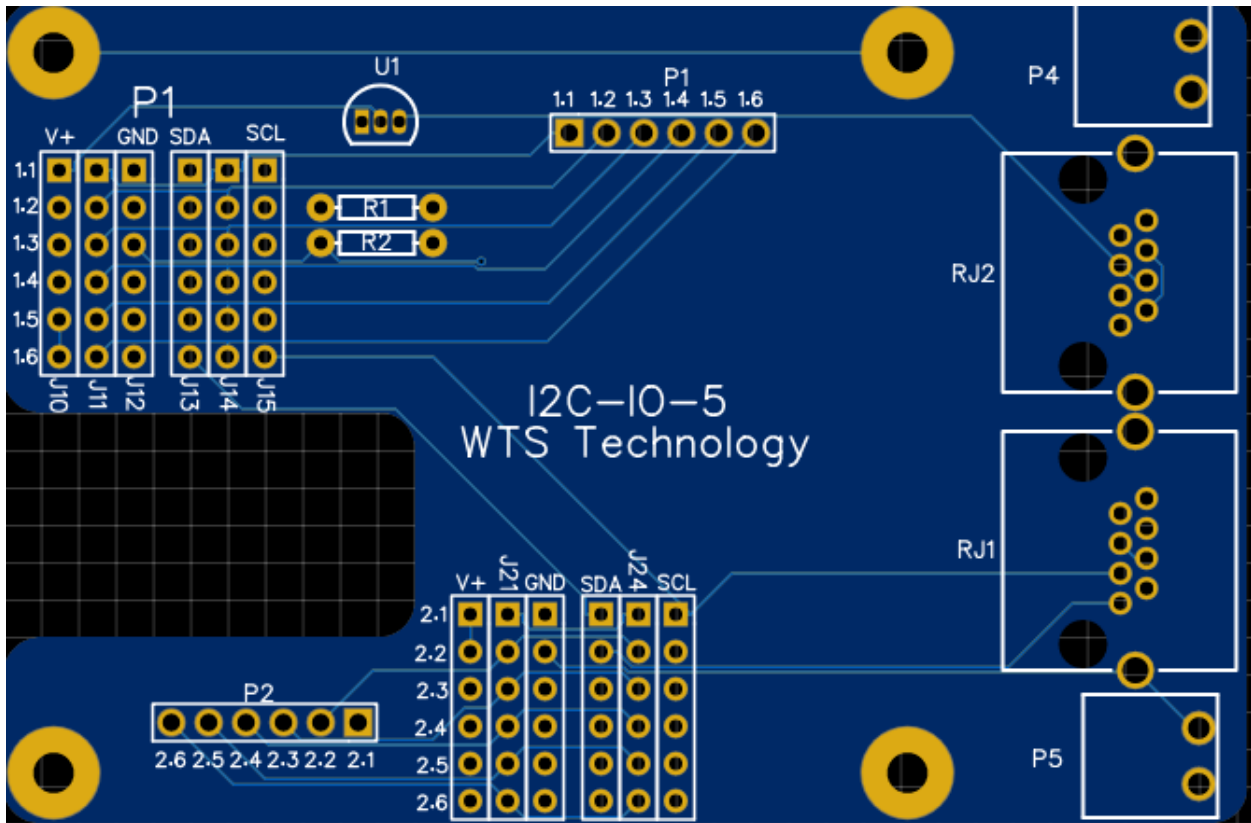
Discontinued, but still available online

DC solid state relay



Jumper setting

I2C I/O Board



Jumper setting

Sensors

BME280

SH1106 OLED



Updated Aug 5, 2021

To do list

1. Test i2c accessory
2. Don't overwrite tmp file if an error occurs once??
3. Fix dew point in wu_upload
4. Add sensor to the water pump
5. Add horn
6. rootfs not expanded??? Watchdog does not use utils.expand_rootfs
7. Write a task to backup the .dat file
8. Connect the air exchanger
9. Write an I2C function to control air exchanger '
10. Finish IP change



Directory structure

/home/pi/automation/	run.py UB_interface.py UB_Utility.py user.dat utils.py watchdog.py
/home/pi/automation/data/	*.dat
/home/pi/automation/docs/	
/home/pi/automation/fonts/	*.ttf
/home/pi/automation/logs/	events.log errors.log
/home/pi/automation/shares/	remote RPi 1 remote RPi 2 etc...
/home/pi/automation/var/	run.var UB_interface.var watchdog.var *.var
/home/pi/automation/var/alarm	*.dis
/home/pi/automation/var/alarm_local	*.dis
/home/pi/automation/var/display	*.dis
/home/pi/automation/var/display_local	*.dis
/home/pi/automation/var/temp	*.tmp
/home/pi/automation/var/temp_local	*.tmp



User data file (user.dat)

- user.dat is located in the main directory (/home/pi/automation). It contains all the information to run the control functions

Global variables

gpio_enb	list of gpio output used for i2c component on the UB board
gpio_io	list of gpio I/O used on the UB board
key_1	key used for the email password
key_2	key used for the RPi password
log_mode	variable used by the "log" function to determined how to do
the log of event / error (default = 1)	
r_email	recipient email address
s_email	sender email address
update_list	contains the list of files to be updated automatically by the function
	update_files (if in use)
wu_url	URL to upload weather data to Weather Underground
wu_user	username for Weather Underground
wu_key	key for Weather Underground

RPi specific variables

192.168.0.xxx_name	common name of the RPi
192.168.0.xxx_ip_address	IP address with subnet mask in short form
192.168.0.xxx_gateway	gateway
192.168.0.xxx_dns	primary DNS server
192.168.0.xxx_hostname	hostname of the device
192.168.0.xxx_workgroup	workgroup of the device

Task variables

192.168.0.xxx_task_x	x = the unique number of the task for this RPi (values 1 to 99)
----------------------	---

GPIO task variables

192.168.xxx.xxx_gpio_task_x	x = the unique number of the I2C task for this RPi (values 1to 99)
-----------------------------	--



List of available functions

BME280

192.168.xxx.xxx_gpio_task_x=i2c_bme280,1,20,Env_Test,1,1;-50;50,12,118

Function to read environmental sensor and prepare the output for multiple file.
Refer to the "bme280" section under URB Interface / Functions

I/O read

192.168.xxx.xxx_gpio_task_x=i2c_io_read ,1,30,Water_Table_Low,0,0;;,12,33,5

Refer to the "i2c_io_read" section under URB Interface / Functions

Latch

192.168.xxx.xxx_gpio_task_x=i2c_latch,1,30,Underground_Water_Pump,Water_Table_Low;ON,Water_Table_High;ON,1,0;0;0,12,33,2

Refer to the "i2c_latch" section under URB Interface / Functions

Manage Disk Space

192.168.0.xxx_task_x=manage_disk_space,1,126,Disk_space,0,1,/media/pi/USBSTICK,500000

Refer to the "manage_disk_space" section under RPI Interface / Functions

Record Camera

192.168.0.xxx_task_x=record_cam,1,120,Front_Cam,rtsp://@192.168.0.9/live4.sdp,/media/pi/USBSTICK,1800

Refer to the "record_cam" section under RPI Interface / Functions

Samba Client

192.168.0.xxx_task_x=samba_client,1,300,samba_client,0,1

Refer to the "samba_client" section under RPI Interface / Functions



SH1106 OLED Display

192.168.0.254_gpio_task_x=i2c_sh1106,1,3,sh1106_display,12,60

Refer to the "i2c_sh1106" section under RPI Interface / Functions

Thermostat

192.168.0.254_gpio_task_x=i2c_thermostat,1,60,Furnace_Heat,Env_EHouse,0,1;10;29,0:00;22,2:00;21,-1.5,21,32,1

Refer to the "i2c_thermostat" section under RPI Interface / Functions

Update files

192.168.0.xxx_task_x=update_files,1,60,update_files,0,1

Refer to the "update_files" section under RPI Interface / Functions

Watchdog

Startup

Create the log directory if it does not exist.

On first boot

Update the RPi

Expand the rootfs partition

Delete the boot1 file

Reboot the RPi

Read the "user.dat" file

Create the "data" directory if it does not exist.

Create the "fonts" directory if it does not exist.

Create the "shares" directory if it does not exist.

Create the "var" directory if it does not exist.

Mount the 'var' directory to RAM

Create the "var/display" directory if it does not exist.

Create the "var/display_local" directory if it does not exist.

Create the "var/temp" directory if it does not exist.

Create the "var/temp_local" directory if it does not exist.

Create the "var/alarm" directory if it does not exist.

Create the "var/alarm_local" directory if it does not exist.



Loop

Write to the watchdog.var file if more than 20 seconds since last write.

Re-load the "user.dat" file if it has changed.

Read the "var/rpi_interface.var" file and restart RPi_interface.py if it hasn't been updated in 60 seconds.

Read the "var/urb_interface.var" file and restart URB_interface.py if it hasn't been updated in 60 seconds.

Functions

None

RPi Interface

Startup

- Writes to the event log file
- Reads the user data file

Loop

- Writes to the "RPi_interface.var" file if it hasn't been updated in the last 30 seconds.
- Gets the IP address of the RPi
- Re-reads the user data file if the file time has changed since startup
- Loads the task list from the user data file.
- Runs the tasks listed in the user data file.

Functions

manage_disk_space(var_data)

The function deletes the oldest file if the storage available space is below value 7. It also logs to the event file the name of the deleted file and terminates by sending data to the write_data function.

var_data = manage_disk_space,ENABLE,120,DISABLE,ENABLE,Disk_space,/media/pi/HD1,50

manage_disk_space	0	function name
,		Separator for main data
ENABLE	1	ENABLE or DISABLE this function from the user.dat file.
,		Separator for main data
120	2	Interval in seconds before re-running the function
,		Separator for main data
DISABLE	3	ENABLE to display on OLED
,		Separator for main data
ENABLE	4	ENABLE or DISABLE alarms for this function
,		Separator for main data
Disk_space	5	Common name of the task. Name can be changed to any name separated by underscore.
,		Separator for main data
/media/pi/HD1	6	Folder / location to manage the disk space
,		Separator for main data
50	7	free disk space required before deleting the oldest file (in Megabytes)

*updated Jun 03, 21

record_cam(var_data)

The function records an rtsp stream in mp4 format to a storage device for a programmable amount of time. Once the elapse time has expired it records a new file. The file name includes the time stamp at the start of the recording.

var_data=record_cam,ENABLE,120,ENABLE,ENABLE,Front_Cam,rtsp://@192.168.0.9/live4.sdp,/media/pi/HD1,1800

record_cam	0	function name
,		Separator for main data
ENABLE	1	ENABLE or DISABLE this function from the user.dat file.
,		Separator for main data
120	2	Interval in seconds before-running the function to verify it is still recording
,		Separator for main data

ENABLE	3	ENABLE to display on OLED
,		Separator for main data
ENABLE	4	ENABLE or DISABLE alarms for this function
,		Separator for main data
Front_Cam	5	Common name of the task. Name can be changed to any name separated by underscore.
,		Separator for main data
rtsp://@192.168.0.9/live4.sdp	6	Address of the rtsp data
,		Separator for main data
/media/pi/USBSTICK	7	location to store the recorded file
,		Separator for main data
1800	8	Length of the recording

*updated Jun 03, 21

samba_client(var_data)

The function will connect shared location from other RPi listed in the user.dat file if they have samba_server running.

var_data=samba_client,ENABLE,300,ENABLE,ENABLE,Samba_Client

samba_client	0	function name
,		Separator for main data
ENABLE	1	ENABLE or DISABLE this function from the user.dat file.
,		Separator for main data
300	2	Interval in seconds before re-running the function
,		Separator for main data
ENABLE	3	ENABLE to display on OLED
,		Separator for main data
ENABLE	4	ENABLE or DISABLE alarms for this function
,		Separator for main data
Samba_Client	5	Common name of the task. Name can be changed to any name separated by underscore.

*updated Jun 03, 21

samba_server(var_data)

The function is used to share the /home/pi/automation directory with other RPi.

var_data=samba_sever,ENABLE,300,ENABLE,ENABLE,Samba_Server

samba_server	0	function name
,		Separator for main data
ENABLE	1	ENABLE or DISABLE this function from the user.dat file.
,		Separator for main data
300	2	Interval in seconds before re-running the function
,		Separator for main data
ENABLE	3	ENABLE data display on OLED
,		Separator for main data
ENABLE	4	ENABLE or DISABLE alarms for this function
,		Separator for main data
Samba_Server	5	Common name of the task. Name can be changed to any name separated by underscore.

*updated Jun 03, 21

test_connectivity(var_data)

The function pings an IP or domain name to verify connectivity.

var_data=test_connectivity,ENABLE,60,ENABLE,ENABLE,Internet,google.ca

test_connectivity	0	function name
,		Separator for main data
ENABLE	1	ENABLE or DISABLE this function from the user.dat file.
,		Separator for main data
60	2	Interval in seconds before running.
,		Separator for main data
ENABLE	3	ENABLE data display on OLED
,		Separator for main data
ENABLE	4	ENABLE or DISABLE alarms for this function
,		Separator for main data
Internet	5	Common name of the task. Name can be changed to any name separated by underscore.
,		Separator for main data
google.ca	6	Domain or IP to test

*updated Jun 04, 21



update_files(var_data)

The function updates files from other RPi listed in the user.dat file. The list of files updated includes the "update_list" from the user data file, the alarm files, the .tmp files and the display files.

var_data=update_files,ENABLE,60,ENABLE,ENABLE,Update_Files

update_files	0	function name
,		Separator for main data
ENABLE	1	ENABLE or DISABLE this function from the user.dat file.
,		Separator for main data
60	2	Interval in seconds before re-running the function
,		Separator for main data
ENABLE	3	ENABLE data display on OLED
,		Separator for main data
ENABLE	4	ENABLE or DISABLE alarms for this function
,		Separator for main data
Update_Files	5	Common name of the task. Name can be changed to any name separated by underscore.

*updated Jun 04, 21

wu_forecast(var_data)

Function to retrieve weather data from Weather Underground API and extract the short forecast for today and tomorrow and save it as display files. Requires an API key in user.dat. The function can generate alarms for high and low temperatures.

var_data=wu_forecast,ENABLE,300,ENABLE,ENABLE;-50;40,WU_Forecast,1800

wu_forecast	0	function name
,		Separator for main data
ENABLE	1	Function ENABLE or DISABLE
,		Separator for main data
300	2	Interval in seconds before re-running the function
,		Separator for main data
ENABLE	3	Enable or disable display to OLED
,		Separator for main data
ENABLE;-50;40	4	Enable or disable alarm. The first value generates an alarm if the temperature goes below and the second value generates an alarm if the value is above



WU_Forecast	5	name of the task and also name of the variable and data file
,		Separator for main data
1800	6	Elapse time between requests to the API

*updated Jun 04, 21

wu_upload(var_data)

Function to upload weather data to the Weather Underground service via HTTP.

var_data=wu_upload,ENABLE,300,ENABLE,ENABLE,WU_Upload,Env_Outdoor,,

wu_upload	0	function name
,		Separator for main data
ENABLE	1	Function ENABLE or DISABLE
,		Separator for main data
300	2	Interval in seconds before re-running the function
,		Separator for main data
ENABLE	3	Enable or disable display to OLED (not in use for this function)
,		Separator for main data
ENABLE	4	Enable or disable alarm.
,		Separator for main data
WU_Upload	5	name of the task and also name of the variable and data file
,		Separator for main data
Env_Outdoor	6	Sensor to read temperature, pressure and humidity from
,		Separator for main data
	7	Future UV sensor name
,		Separator for main data
	8	Future wind data sensor name



URB Interface

Startup

- Read the user.dat file
- Get the list of i2c enable pins
- Setup i2c enable pins as output with value 0
- Get the list of GPIO input / output pins
- Setup the GPIO input / output pins as input

Loop

- Write to the "var/ub.var" file if more than 30 seconds since last write.
- Reload the "user dat" file if it has changed or the IP has changed.
- Load the gpio task list.
- Skip the task if the task flag is 0 and the var file does not exist.
- Run the task if it hasn't been run for more than interval time or the var file does not exist.
- Run i2c_detect if var/i2c_detect.var exist.

Functions

i2c_accessory(var_data)

I2C function that uses multiple input to control 1 output. Inputs are .tmp files stored in RAM. There are no maximum inputs.

Note : The logic is done in sequentially meaning input 1 and 2, then the result of 1 and 2 with input 3, etc.

var_data=i2c_accessory,ENABLE,30,ENABLE,ENABLE,EHouse_Heat_Fan,Radian_Heat_Garage;OFF&Radian_Heat_Solarium;OFF&WHouse_Heat;OFF&Radian_Liquid_Level;ON,18,32,1

i2c_accessory	0	Function name
ENABLE	1	Sensor ENABLE or DISABLE
30	2	Interval in seconds before re-running the function
ENABLE	3	Enable or disable display to OLED
ENABLE	4	Enable or disable alarm.
EHouse_Heat_Fan	5	Name / Location of sensor (Separate words with under score)
Radian_Heat_Garage;OFF&Radian_Heat_Solarium;OFF&WHouse_Heat;OFF&Radian_Liquid_Level;ON	6	Conditions for controlling the output. Each condition must be separated by '&' (AND) or ' ' (OR) and the status of each condition is separated by ';'.
18	7	Pin to enable on the URB board (GP1=18,GP2=4,GP3=22,GP4=13,GP5=21,GP6=12)
32	8	Address of the PCF8574 I/O board. (32,33,34,35,56,57,58,59)
1	9	Bit of the the I2C pcf8574 to write (1,2,3,4)

Note: Data must be separated by comma

Updated Jun 17, 2021

i2c_bme280(var_data)

I2C function to read the BME280 environmental sensor.

var_data = i2c_bme280,ENABLE,60,ENABLE,ENABLE;-50;50,Env_Test,22,118

i2c_bme280	0	Function name
ENABLE	1	Sensor ENABLE or DISABLE
60	2	Interval in seconds before re-running the function
ENABLE	3	Enable or disable display to OLED
ENABLE;-50;50	4	Enable or disable alarm. The first value generates an alarm if the temperature goes below and the second value generates an alarm if the value is above.
Env_Test	5	Name / Location of sensor (Separate words with under slash)
22	6	pin to enable on the UB Interface board (GP1=18,GP2=4,GP3=22,GP4=13,GP5=21,GP6=12)
118	7	Decimal address of the sensor (118, 119)

Note: Data must be separated by comma

**updated Jun 04, 21*

i2c_detect()

I2C function used by UB Utility to detect I2C devices from the 6 port of the UB board.

i2c_io_read(var_data)

I2C function used to read input from connected I2C I/O board. The output data is stored in RAM via the "write_data" function.

var_data=i2c_io_read,ENABLE,5,ENABLE,ENABLE;OFF,Input_Test,22,33,5

i2c_io_read	0	function name
,		Separator for main data
ENABLE	1	Read status of pin ENABLE or DISABLE
,		Separator for main data
5	2	Interval in seconds before re-running the function
,		Separator for main data
ENABLE	3	Enable or disable display to OLED
,		Separator for main data
ENABLE;OFF	4	Enable or disable alarm. If the value = the first value an alarm will be generated

,		Separator for main data
Input_Test	5	Input pin to read
,		Separator for main data
22	6	pin to enable on the UB Interface board (GP1=18,GP2=4,GP3=22,GP4=13,GP5=21,GP6=12)
,		Separator for main data
33	7	decimal address of the sensor (118, 119)
		Separator for main data
5	8	bit

*updated Jun 05, 21

i2c_latch(var_data)

I2C function that reads 2 input from ".tmp" files and that generate a latch output based on the condition of the inputs. The output is generated for a I2C I/O Board and the data is outputted to the "write_data" function.

var_data=i2c_latch,ENABLE,30,ENABLE,ENABLE,Underground_Water_Pump,Water_Table_Low;ON,Water_Table_High;ON,12,33,2

i2c_latch	0	function name
,		Separator for main data
ENABLE	1	ENABLE or DISABLE to available OLED display
,		Separator for main data
30	2	Interval in seconds before re-running the function
,		Separator for main data
ENABLE	3	Enable or disable display to OLED
,		Separator for main data
ENABLE	4	Enable or disable alarm. If the value = the first value an alarm will be generated
,		Separator for main data
Underground_Water_Pump	5	Name / Location of the device to be controlled. (Separate words with under slash)
,		Separator for main data
Water_Table_Low;ON	6	Sensor to turn "OFF" the latch
,		Separator for main data
Water_Table_High;ON	7	Sensor to turn "ON" the latch
,		Separator for main data
12	8	pin to enable on the UB Interface board

		(GP1=18,GP2=4,GP3=22,GP4=13,GP5=21,GP6=12)
,		Separator for main data
33	9	decimal address of the sensor
,		Separator for main data
2	10	Bit to enable (1 to 4)

Errors

- FILE READ ERROR - tmp file for one of the 2 inputs do not exist
- INPUT ERROR - tmp file for one of the 2 inputs has an error value ('ERR')
- CHIP READ ERROR - Unable to read the PCF8574

i2c_pcf8574(enable_pin,address,bit,bit_x)

to be replaced*****

I2C function used to read / write to the PCF8574 on an I2C I/O Board. The function returns 'ON', "OFF" or "ERR".

enable_pin	(GP1=18,GP2=4,GP3=22,GP4=13,GP5=21,GP6=12)
address	Decimal values (32 to 39 or 56 to 63)
bit	bit to read or change
bit_x	"R" - Read "ON" - change bit value to 0 "OFF" change bit value to 1

i2c_pcf8574_read(enable_pin,address,bit)

Function to read (1) bit of a PCF8574.



enable_pin	(GP1=18,GP2=4,GP3=22,GP4=13,GP5=21,GP6=12)
address	Decimal values (32 to 39 or 56 to 63)
bit	bit to read or change

The returned value is either ON, OFF or ERR. ON for low input, OFF for high input and ERR for an error reading the pin.

`i2c_pcf8574_write(enable_pin,address,bit,value)`

Function to write to (1) pin of a PCF8574.

enable_pin	(GP1=18,GP2=4,GP3=22,GP4=13,GP5=21,GP6=12)
address	Decimal values (32 to 39 or 56 to 63)
bit	Bit to write to. Bit 1 to 4 only as bit 5 to 8 are input
value	ON or OFF

The returned value is either ON, OFF or ERR. ON for low input, OFF for high input and ERR for an error writing to the pin or to the chip.

`i2c_pins_disable()`

The function is used to disable all enabled pins of the URB board. Enable pins are GP1=18,GP2=4,GP3=22,GP4=13,GP5=21,GP6=12 to be removed



i2c_sh1106(var_data)

I2C function used to convert display files and them to a SH1106 OLED display.

var_data=i2c_sh1106,ENABLE,2.5,DISABLE,ENABLE,SH1106_Display,12,60

i2c_sh1106	0	function name
,		Separator for main data
ENABLE	1	ENABLE or DISABLE to available OLED display
,		Separator for main data
2.5	2	Interval in seconds before re-running the function
,		Separator for main data
DISABLE	3	Enable or disable display to OLED (not in use for this function)
,		Separator for main data
ENABLE	4	Enable or disable alarm.
,		Separator for main data
SH1106_Display	5	Name / Location of sensor (Separate words with under score)
,		Separator for main data
12	6	pin to enable on the UB Interface board (GP1=18,GP2=4,GP3=22,GP4=13,GP5=21,GP6=12)
,		Separator for main data
60	7	decimal address of the OLED display (60)

*updated Jun 04, 21

i2c_timer(var_data)

I2C function to control (1) output based on the time of the day. There is no maximum of time entry but the entries have to be in order using 24 hours format.

var_data=i2c_timer,ENABLE,5,ENABLE,ENABLE,Test_Timer,6:00.OFF&15:00;ON&20:00;OFF&22:02;ON,22,33,1

i2c_timer	0	function name
,		Separator for main data
ENABLE	1	Read status of pin ENABLE or DISABLE
,		Separator for main data
5	2	Interval in seconds before re-running the function
,		Separator for main data
ENABLE	3	Enable or disable display to OLED
,		Separator for main data
ENABLE	4	Enable or disable alarm.
,		Separator for main data
Test_Timer	5	
,		Separator for main data
6:00.OFF&15:00;ON&20:00;OFF&22:02;ON	6	
,		Separator for main data
22	7	pin to enable on the UB Interface board (GP1=18,GP2=4,GP3=22,GP4=13,GP5=21,GP6=12)
,		Separator for main data
33	8	
		Separator for main data
1	9	bit

i2c_thermostat(var_data)

I2C function to control I2C I/O Board output for temperature regulation. The function can be used for heating or cooling and has a programmable temperature variation.

var_data=i2c_thermostat,ENABLE,4,ENABLE,ENABLE;-50;50,Test_Thermostat, Env_Test, 6:00;27;13:00;20,7:00;26;14:00;20,-1.5,22,33,2

i2c_thermostat	0	function name
,		Separator for main data
ENABLE	1	Thermostat is ENABLE or DISABLE
,		Separator for main data
5	2	Interval in seconds before re-running the function
,		Separator for main data
ENABLE	3	Enable or disable display to OLED
,		Separator for main data
ENABLE;-50;50	4	Enable or disable alarm. The first value generates an alarm if the temperature goes below and the second value generates an alarm if the value is above
Test_Thermostat	5	Name / Location of the device to be controlled. (Separate words with underscore)
,		Separator for main data
Env_Test	6	Sensor to be used to control the temperature
,		Separator for main data
6:00;27&13:00;20	7	Weekday Schedule start time and temperature (Multiple time/temperature can be added separated by "&")
,		Separator for main data,
7:00;26&14:00;20	8	Weekend Schedule start time and temperature (Multiple time/temperature can be added separated by "&")
,		Separator for main data
-1.5	9	Total temperature variation. (+/- half of value). Use positive value for heating and negative value for cooling.
,		Separator for main data
22	10	pin to enable on the UB Interface board (GP1=18,GP2=4,GP3=22,GP4=13,GP5=21,GP6=12)
,		Separator for main data
33	11	decimal address of the sensor
,		Separator for main data
2	12	Bit to enable (1 to 4)



*updated Jun 05, 21

Utility file (utils.py)

Functions

`bash_command(bash_input)`

Function to execute bash command and return its output and an exit code. This function uses `subprocess.check_output` and uses a different syntax for the command which does not require using commas. It return `exit_code = 0` if successful + data generated

`dumb_code(a, b)`

Encode / decode function.. Encode b if a = 0. Decode b if a = 1.`code_key(data, selection)`

`file_exists(path_file)`

Function to verify if a path or a file exists. Return and exit code = 0 if the file exists and exit code = 1 if the file is not found. Requires `bash_command`.

`file_get_mod_time(path_file)`

Function to get the last modification time of the file. If the file is not found it returns a file time of 0. Requires `os.path.getmtime` and `file_exists`

`file_read(path_file)`

Function to read the content of a file. The function will try to re-read the file for 1 sec if it is empty. This was necessary so that when a file is being written to, that the read output is not empty. Returns exit code of 0 and the data read, if successful. Requires `file_get_mode_time`.

`file_write(path_file)`

Function to overwrite a file. The function will try to write to the file for 1 second. This is necessary when multiple functions are writing to the same file. Returns an exit code of 0 if successful.

`i2c_device()`

Returns the list of i2c devices available on the RPi / UB board.



log(data_text, event_file, log_mode)

Function to log / print to screen, events and errors with timestamp. If log_mode = 0 the function will only print to the terminal screen. If log_mode = 2 the function will only be written to the text file. If log_mode = 1 it will do both. During the writing to file, the file/directory will be created if it doesn't exist. The newly written text will be at the top of the file. An exit code of 0 is returned if successful. Requires time, time.localtime, time.strftime, file_read, file_write.

lookup_text(x_data, text)

Function to find text in a variable. The function returns an exit code of 0 if the value is found.

lookup_var(x_data, x_search)

Function to lookup x_search string in x_data variable. The function is used to retrieve variables after an "=" sign and before the EOL. The result is stripped of any preceding or following blank character. Returns an exit code = 0 if successful, exit code of 1 if x_search not found.

pid_util(pid_name, selection)

Function to verify if a linux process running by finding its PID. It can also terminate the pid_name. Requires bash_command. Return 0 if

pid_util	function name
pid_name	Name of the running program. Partial names can be used
selection	0 = verify if the program is running, 1 terminate the program
	Returns 0 if the PID is running or as been terminated and returns 1 if the PID is not running.

send_zoho_email(subject, text, event_file, error_file, log_mode)

Function to send email via the ZOHO SMTP server. The function writes to the event file if successful or to the error file if it fails to send the email.

signed_short(msb, lsb)

Function used by the BME 280 Environmental Sensor.

write_data(data):

Function used to write data to the data files, alarm files, temp files and display files.

data[0] = status	ENABLE or DISABLE. Dependent of value set in the user.dat file. DISABLE - Delete var, tmp, display and alarm files of the task_name if they exist on the local and remote RPi. ENABLE - Delete alarm file if it exists and there are no current alarms, ie data[4] = blank - Store the new value to the tmp file if the value has changed. If there is an alarm add an '*' to the end of the string. - Store the new value in the data file if the value has changed. - Log to the event file if the value has changed. - Update the display file if the data[3] is not blank. - Delete the display file if data[3] is blank and a display file exist. - Writes to the .var file
data[1] = task_name	Name use for creating / updating / deleting variable files
data[2] = tmp_data	String of data; for the tmp file and data file. Contains only one value + an '*' if there is an error or alarm on one of the inputs.
data[3] = display_data	String of data; for the display file if the display flag in the user.dat file is set to ENABLE. String may contain 1 or 2 values separated by ','
data[4] = alarm_data	String of data; for the alarm file if an alarm exists and the alarm flag in the user.dat file is set to ENABLE. String may contain 1 or 2 values separated by ','
data[5] = log_mode	The type of logging (sourced from the user.dat)

Old file

- ms4448,1,120,MS4448
 - ms4448 function name
 - 1 status (inactive = 0, active = 1)
 - 120 interval in seconds before re-running the function
 - MS4448 interval variable (var/MS4448.var)



- ms4448_display,1,120,MS4448_display
 ms4448_display function name
 1 status (inactive = 0, active = 1)
 120 Interval in seconds before re-running the function
 MS4448_display interval variable (var/MS4448_display.var)
- wu_forecast,1,120,WU_Forecast
 wu_forecast function name
 1 status (inactive = 0, active = 1)
 120 Interval in seconds before re-running the function
 WU_Forecast Interval variable (var/WU_Forecast.var)
- wu_upload,1,300,WUupload,Outdoor
 wu_upload function name
 1 status (inactive = 0, active = 1)
 300 Interval in seconds before re-running the function
 WUupload interval variable (var/WUupload.var)
 Outdoor name of the BME 280 sensor to use

i2c_tsl2561

i2c_tsl2561,1,30,Lux_Meter,1,0,18,56

i2c_tsl2561	function name
,	Separator for main data
1	status (inactive = 0, active = 1)
,	Separator for main data
30	Interval in seconds before re-running the function
,	Separator for main data
Lux_Meter	input to read (Separate words with under slash)
,	Separator for main data
1	Display flag (0=don't display, 1=display, 2=display if ON, 3=display if OFF)
,	Separator for main data
0	Alarm flag (0=don't generate alarm, 2=generate alarm if ON, 3=Generate alarm if OFF)
,	Separator for main data
18	pin to enable when using RPi board V6 or higher (GP1=18,GP2=4,GP3=22,GP4=13,GP5=21,GP6=12)
,	Separator for main data
57	address of the the I2C pcf8574 (57)

Functions

- `ms4448()`

Function to start the ms4448x function in a separate window.

- `ms4448x()`

Function to read data from a Magnum MS4448 Inverter and PT-100 charge controller. The data is stored in the variable `var_dir/solar.var`

- `ms4448_display()`

Function to start the `ms4448_display_x` function in a separate window.

- `ms4448_display_x()`

Function to display the data stored in `var_dir/solar.var`.

- `wu_upload()`

Function to upload temperature, humidity, pressure to Weather Underground.

- `add_samba_pwd()`

Function to add samba password automatically. This function is only used on the first boot after writing a new SD card.

Return `exit_code = 0` if successful

Requires `subprocess.Popen` and `PIPE`

- `text_change(x_data, old_text, new_text)`

Function to change text in a variable. Non zero exit code if the text is not found.

Returns exit code and modified variable.

Requires `lookup_text`

- `whoami()`

Function that returns the current user and an exit code.