



In This document We will see how to design the Window Layout and Send Content to the LED Board. We will use the Raw Packet Format Sent over a serial port. We will later introduce how we can use the same packet over REST Web Services API. There are helper functions for JS and other languages that make creating these packets more developer friendly.

The Protocol Consists of

- Start of Packet Character '[',
- Board ID - 01,
- Command - T,
- Parameters for the command and Text to Display.
- CRC (if CRC is XX then CRC checking is skipped.)
- End of packet]

[01T01000104300202ABCDEFGHJKLMNOPQRSTXX]

Start Of Packet [
End Of Packet]
Board ID 01
CRC XX

Setup Window

The first thing to set up would be the Window Layout , You can have up to 16 Windows at any given time with each window having independently updatable content.

[01W02,
0,0,32,16,
32,0,32,16,XX]

Ex.	LEN	Description
[1	SOP

01	2	ID
W/V	1	Window Set Command . If you want to save the window parameter use V command.
02	2	Window Count (00-15)
0	CommaSep	X1
0	CommaSep	Y1
32	CommaSep	W1
16	CommaSep	H1
32	CommaSep	X2
0	CommaSep	Y2
32	CommaSep	W2
16	CommaSep	H2
XX	2	CRC
]	1	End of Packet

Send Messages To Window

This packet allows one to Send messages.

```
// TTWWFFEESPSTAC
[01T01000104300202abcdefghijklmnopqrstuvwxyXX]
```

Value	LEN	Description
[1	SOP
01	2	ID
T	1	Set Message to Window
00	2	Window Type. 0 : WIN_TYPE_NOINIT 1 : WIN_TYPE_MESSAGE

		<p>2 : WIN_TYPE_TEMPERATURE_C 3 : WIN_TYPE_HUMIDITY 4 : WIN_TYPE_GPIO_COUNTER 5 : WIN_TYPE_GPIO_HILOW_MESSAGE 6 : WIN_TYPE_GPIO_BIN_MESSAGELOOKUP 7 : WIN_TYPE_GPIO_MESSAGELOOKUP 8 : WIN_TYPE_TEMPERATURE_F 11 : WIN_TYPE_MODBUS 12 : WIN_TYPE_MODBUS_MSGLOOKUP 14 : WIN_TYPE_MODBUS_TRANSFORM 15 : WIN_TYPE_MODBUS_TRANSFORM_RV 21 : WIN_TYPE_TIME_HHMM_24 22 : WIN_TYPE_TIME_HHMMSS_24 23 : WIN_TYPE_TIME_HHMM_12 24 : WIN_TYPE_TIME_HHMMSS_12 25 : WIN_TYPE_TIME_DDMMYY 26 : WIN_TYPE_TIME_DDMMYYYY 27 : WIN_TYPE_TIME_FORMAT 30 : WIN_TYPE_REMOTE_ENTRY 31 : WIN_TYPE_REMOTE_ENTRY_FORMAT 40 : WIN_TYPE_SUM 41 : WIN_TYPE_DIFF 42 : WIN_TYPE_MUL 43 : WIN_TYPE_DIV 44 : WIN_TYPE_EFF 45 : WIN_TYPE_TRANSFORM 46 : WIN_TYPE_TRANSFORM_RV 50 : WIN_TYPE_TIMER_MMSS 51 : WIN_TYPE_TIMER_HHMM 52 : WIN_TYPE_TIMER_HHMMSS 60 : WIN_TYPE_GIF</p> <p>99 : WIN_TYPE_EMPTY 99 : MAX_WTYPE</p>
00	2	Window To send the Message.
01	2	<p>Font</p> <p>01 : FONT_5X7 02 : FONT_8 03 : FONT_16 04 : FONT_16_WIDE 05 : FONT_24 06 : FONT_16_LARGE_NUM 07 : FONT_28 08 : FONT_32 09 : FONT_32N</p>

04	2	<p>Effect</p> <ul style="list-style-type: none"> 0 : SCROLL_NODATA 1 : SCROLL_ANIM_STAY 2 : SCROLL_ANIM_SCROLL_LEFT 3 : SCROLL_ANIM_SCROLL_CONT_LEFT 4 : SCROLL_ANIM_SCROLL_BLINK_LEFT 5 : SCROLL_ANIM_BLINK 6 : SCROLL_ANIM_BOT2TOP 7 : SCROLL_ANIM_CONT_BOT2TOP 8 : SCROLL_ANIM_BLINK_BOT2TOP 9 : SCROLL_ANIM_WIPE_TOP2BOT 10 : SCROLL_ANIM_WIPE_BLINK_TOP2BOT 11 : SCROLL_ANIM_WIPE_BOT2TOP 12 : SCROLL_ANIM_WIPE_BLINK_BOT2TOP 13 : SCROLL_ANIM_WIPE_L2R 14 : SCROLL_ANIM_WIPE_BLINK_L2R 15 : SCROLL_ANIM_WIPE_R2L 16 : SCROLL_ANIM_WIPE_BLINK_R2L 17 : SCROLL_ANIM_WIPE_TOP2BOT_FULL 18 : SCROLL_ANIM_WIPE_BLINK_TOP2BOT_FULL 19 : SCROLL_ANIM_WIPE_BOT2TOP_FULL 20 : SCROLL_ANIM_WIPE_BLINK_BOT2TOP_FULL 21 : SCROLL_ANIM_WIPE_L2R_FULL 22 : SCROLL_ANIM_WIPE_BLINK_L2R_FULL 23 : SCROLL_ANIM_WIPE_R2L_FULL 24 : SCROLL_ANIM_WIPE_BLINK_R2L_FULL 25 : SCROLL_FIXED_MESSAGE 26 : SCROLL_FIXED_BLINK
32	2	Speed 00-99
16	2	Stay 00-99
1	1	<p>Alignment - 8 bits 0000 HHVV</p> <p>HH - Horizontal Align</p> <ul style="list-style-type: none"> 00 - 0 None 01 - 1 Left 10 - 2 Center 11 - 3 Right <p>VV - Vertical Align</p> <ul style="list-style-type: none"> 00 - 0 01 - 1 Top

		10 - 2 Center 11 - 3 Bottom																																																
1	1	Color 0x0-0xF (15) <table border="1"> <tr><td>0</td><td>■</td><td>#000000</td></tr> <tr><td>1</td><td>■</td><td>#FF0000</td></tr> <tr><td>2</td><td>■</td><td>#00FF00</td></tr> <tr><td>3</td><td>■</td><td>#FFFF00</td></tr> <tr><td>4</td><td>■</td><td>#0000FF</td></tr> <tr><td>5</td><td>■</td><td>#FF00FF</td></tr> <tr><td>6</td><td>■</td><td>#00FFFF</td></tr> <tr><td>7</td><td>■</td><td>#FFFFFF</td></tr> <tr><td>8</td><td>■</td><td>#000080</td></tr> <tr><td>9</td><td>■</td><td>#008000</td></tr> <tr><td>A</td><td>■</td><td>#008080</td></tr> <tr><td>B</td><td>■</td><td>#800000</td></tr> <tr><td>C</td><td>■</td><td>#800080</td></tr> <tr><td>D</td><td>■</td><td>#808000</td></tr> <tr><td>E</td><td>■</td><td>#C0C0C0</td></tr> <tr><td>F</td><td>■</td><td>#808080</td></tr> </table>	0	■	#000000	1	■	#FF0000	2	■	#00FF00	3	■	#FFFF00	4	■	#0000FF	5	■	#FF00FF	6	■	#00FFFF	7	■	#FFFFFF	8	■	#000080	9	■	#008000	A	■	#008080	B	■	#800000	C	■	#800080	D	■	#808000	E	■	#C0C0C0	F	■	#808080
0	■	#000000																																																
1	■	#FF0000																																																
2	■	#00FF00																																																
3	■	#FFFF00																																																
4	■	#0000FF																																																
5	■	#FF00FF																																																
6	■	#00FFFF																																																
7	■	#FFFFFF																																																
8	■	#000080																																																
9	■	#008000																																																
A	■	#008080																																																
B	■	#800000																																																
C	■	#800080																																																
D	■	#808000																																																
E	■	#C0C0C0																																																
F	■	#808080																																																
Data	Variable	Text to display																																																
XX	2	CRC																																																
]	1	End of Packet																																																

Replace Text In a Window

This packet allows one to send a message to a Window.

// WWC

[01P002abcdefghijklmnopqrstuvwxyzXX]

Value	LEN	Description
[1	SOP
01	2	ID
P	1	RePlace Message in Window

00	2	Window To Replace the Message.																																																
1	1	<p>Color 0x0-0xF (15)</p> <table border="1"> <tr><td>0</td><td>█</td><td>#000000</td></tr> <tr><td>1</td><td>█</td><td>#FF0000</td></tr> <tr><td>2</td><td>█</td><td>#00FF00</td></tr> <tr><td>3</td><td>█</td><td>#FFFF00</td></tr> <tr><td>4</td><td>█</td><td>#0000FF</td></tr> <tr><td>5</td><td>█</td><td>#FF00FF</td></tr> <tr><td>6</td><td>█</td><td>#00FFFF</td></tr> <tr><td>7</td><td>█</td><td>#FFFFFF</td></tr> <tr><td>8</td><td>█</td><td>#000080</td></tr> <tr><td>9</td><td>█</td><td>#008000</td></tr> <tr><td>A</td><td>█</td><td>#008080</td></tr> <tr><td>B</td><td>█</td><td>#800000</td></tr> <tr><td>C</td><td>█</td><td>#800080</td></tr> <tr><td>D</td><td>█</td><td>#808000</td></tr> <tr><td>E</td><td>█</td><td>#C0C0C0</td></tr> <tr><td>F</td><td>█</td><td>#808080</td></tr> </table>	0	█	#000000	1	█	#FF0000	2	█	#00FF00	3	█	#FFFF00	4	█	#0000FF	5	█	#FF00FF	6	█	#00FFFF	7	█	#FFFFFF	8	█	#000080	9	█	#008000	A	█	#008080	B	█	#800000	C	█	#800080	D	█	#808000	E	█	#C0C0C0	F	█	#808080
0	█	#000000																																																
1	█	#FF0000																																																
2	█	#00FF00																																																
3	█	#FFFF00																																																
4	█	#0000FF																																																
5	█	#FF00FF																																																
6	█	#00FFFF																																																
7	█	#FFFFFF																																																
8	█	#000080																																																
9	█	#008000																																																
A	█	#008080																																																
B	█	#800000																																																
C	█	#800080																																																
D	█	#808000																																																
E	█	#C0C0C0																																																
F	█	#808080																																																
Data	Variable	Text to display																																																
XX	2	CRC																																																
]	1	End of Packet																																																

Saved Messages Templates

Flink Sign has Message Templates that can be loaded into a Window by default when the board boots up. By default Saved Messages 0..15 Correspond are loaded automatically to Windows 0..15

Save Message.

This packet allows one to Save Message.

```
//  TTMMFFEESPSTAC
[01U01000104300202abcdefghijklmnopqrstuvwxyXX]
```

Value	LEN	Description
[1	SOP
01	2	ID
U	1	Save Message
00	2	Message Type. 0 : WIN_TYPE_NOINIT 1 : WIN_TYPE_MESSAGE 2 : WIN_TYPE_TEMPERATURE_C 3 : WIN_TYPE_HUMIDITY 4 : WIN_TYPE_GPIO_COUNTER 5 : WIN_TYPE_GPIO_HILOW_MESSAGE 6 : WIN_TYPE_GPIO_BIN_MESSAGELOOKUP 7 : WIN_TYPE_GPIO_MESSAGELOOKUP 8 : WIN_TYPE_TEMPERATURE_F 11 : WIN_TYPE_MODBUS 12 : WIN_TYPE_MODBUS_MSGLOOKUP 14 : WIN_TYPE_MODBUS_TRANSFORM 15 : WIN_TYPE_MODBUS_TRANSFORM_RV 21 : WIN_TYPE_TIME_HHMM_24 22 : WIN_TYPE_TIME_HHMMSS_24 23 : WIN_TYPE_TIME_HHMM_12 24 : WIN_TYPE_TIME_HHMMSS_12 25 : WIN_TYPE_TIME_DDMMYY 26 : WIN_TYPE_TIME_DDMMYYYY

		<p>27 : WIN_TYPE_TIME_FORMAT 30 : WIN_TYPE_REMOTE_ENTRY 31 : WIN_TYPE_REMOTE_ENTRY_FORMAT 40 : WIN_TYPE_SUM 41 : WIN_TYPE_DIFF 42 : WIN_TYPE_MUL 43 : WIN_TYPE_DIV 44 : WIN_TYPE_EFF 45 : WIN_TYPE_TRANSFORM 46 : WIN_TYPE_TRANSFORM_RV 50 : WIN_TYPE_TIMER_MMSS 51 : WIN_TYPE_TIMER_HHMM 52 : WIN_TYPE_TIMER_HHMMSS 60 : WIN_TYPE_GIF</p> <p>99 : WIN_TYPE_EMPTY 99 : MAX_WTYPE</p>
00	2	<p>MessageID. 00-15 Default Window Messages 16-63 Custom Messages that can be loaded to a window using the load Message To Window Command Ex. [01L0011XX].</p>
01	2	<p>Font 01 : FONT_5X7 02 : FONT_8 03 : FONT_16 04 : FONT_16_WIDE 05 : FONT_24 06 : FONT_16_LARGE_NUM 07 : FONT_28 08 : FONT_32 09 : FONT_32N</p>
04	2	<p>Effect 0 : SCROLL_NODATA 1 : SCROLL_ANIM_STAY 2 : SCROLL_ANIM_SCROLL_LEFT 3 : SCROLL_ANIM_SCROLL_CONT_LEFT 4 : SCROLL_ANIM_SCROLL_BLINK_LEFT 5 : SCROLL_ANIM_BLINK 6 : SCROLL_ANIM_BOT2TOP 7 : SCROLL_ANIM_CONT_BOT2TOP 8 : SCROLL_ANIM_BLINK_BOT2TOP 9 : SCROLL_ANIM_WIPE_TOP2BOT 10 : SCROLL_ANIM_WIPE_BLINK_TOP2BOT 11 : SCROLL_ANIM_WIPE_BOT2TOP</p>

		12 : SCROLL_ANIM_WIPE_BLINK_BOT2TOP 13 : SCROLL_ANIM_WIPE_L2R 14 : SCROLL_ANIM_WIPE_BLINK_L2R 15 : SCROLL_ANIM_WIPE_R2L 16 : SCROLL_ANIM_WIPE_BLINK_R2L 17 : SCROLL_ANIM_WIPE_TOP2BOT_FULL 18 : SCROLL_ANIM_WIPE_BLINK_TOP2BOT_FULL 19 : SCROLL_ANIM_WIPE_BOT2TOP_FULL 20 : SCROLL_ANIM_WIPE_BLINK_BOT2TOP_FULL 21 : SCROLL_ANIM_WIPE_L2R_FULL 22 : SCROLL_ANIM_WIPE_BLINK_L2R_FULL 23 : SCROLL_ANIM_WIPE_R2L_FULL 24 : SCROLL_ANIM_WIPE_BLINK_R2L_FULL 25 : SCROLL_FIXED_MESSAGE 26 : SCROLL_FIXED_BLINK
32	2	Speed 00-99
16	2	Stay 00-99
1	1	Alignment - 8 bits 0000 HHVV HH - Horizontal Align 00 - 0 None 01 - 1 Left 10 - 2 Center 11 - 3 Right VV - Vertical Align 00 - 0 01 - 1 Top 10 - 2 Center 11 - 3 Bottom
1	1	Color 0x0-0xF (15)

		<table border="1"> <tr><td>0</td><td>■</td><td>#000000</td></tr> <tr><td>1</td><td>■</td><td>#FF0000</td></tr> <tr><td>2</td><td>■</td><td>#00FF00</td></tr> <tr><td>3</td><td>■</td><td>#FFFF00</td></tr> <tr><td>4</td><td>■</td><td>#0000FF</td></tr> <tr><td>5</td><td>■</td><td>#FF00FF</td></tr> <tr><td>6</td><td>■</td><td>#00FFFF</td></tr> <tr><td>7</td><td>■</td><td>#FFFFFF</td></tr> <tr><td>8</td><td>■</td><td>#000080</td></tr> <tr><td>9</td><td>■</td><td>#008000</td></tr> <tr><td>A</td><td>■</td><td>#008080</td></tr> <tr><td>B</td><td>■</td><td>#800000</td></tr> <tr><td>C</td><td>■</td><td>#800080</td></tr> <tr><td>D</td><td>■</td><td>#808000</td></tr> <tr><td>E</td><td>■</td><td>#C0C0C0</td></tr> <tr><td>F</td><td>■</td><td>#808080</td></tr> </table> <p>You can also change the Palette Colors using the Set Palette Color Command.</p> <p>Set Color pallete. [IDZC I - Color Index 0-F XXXX - 565 Color in Hex Format 1234 CRC] [01ZC11234XX]</p>	0	■	#000000	1	■	#FF0000	2	■	#00FF00	3	■	#FFFF00	4	■	#0000FF	5	■	#FF00FF	6	■	#00FFFF	7	■	#FFFFFF	8	■	#000080	9	■	#008000	A	■	#008080	B	■	#800000	C	■	#800080	D	■	#808000	E	■	#C0C0C0	F	■	#808080
0	■	#000000																																																
1	■	#FF0000																																																
2	■	#00FF00																																																
3	■	#FFFF00																																																
4	■	#0000FF																																																
5	■	#FF00FF																																																
6	■	#00FFFF																																																
7	■	#FFFFFF																																																
8	■	#000080																																																
9	■	#008000																																																
A	■	#008080																																																
B	■	#800000																																																
C	■	#800080																																																
D	■	#808000																																																
E	■	#C0C0C0																																																
F	■	#808080																																																
Data	Variable	Text to display / Data for the template.																																																
XX	2	CRC																																																
]	1	End of Packet																																																

Load Message Template to a Window

This packet allows one to Load a Message Template to a Window.

```
// WWMM
[01L0001XX]
```

Value	LEN	Description
-------	-----	-------------

[1	SOP
01	2	ID
L	1	Load Message Template to Window
00	2	Window To Replace the Message.
01	2	Message Template ID
XX	2	CRC
]	1	End of Packet

Setting Up Clock.

1) Standard 24 Hour Clock

```
WIN_TYPE_TIME_HHMM_24 = 21
// TMMFFEESPSTAC
[01U21010301000001XX]
// One Window 32x16 with Template 1.
[01W01,1,0,0,32,16XX]
```

2) World Clock with 2 Times.

For World Clock you have to set the Internal clock to UTC and set the Offset for the time windows.

```
Set Time
    HHMMSS
[01S213600XX]
SetDate
    DDMMYY
[01D010722XX]
```

```
WIN_TYPE_TIME_HHMM_24 = 21
// TMMFFEESPSTAC
    @ -/+ Time Offset
    HHMM
[01U21010301000001-0700XX] //PST
[01U21020301000001+0530XX] //IST
// Two Window 32x16 with Template 1,2.
[01W02,1,0,0,32,16,2,0,16,32,16XX]
```

The same applies for the Following WindowTemplates.

WIN_TYPE_TIME_HHMM_24	21
WIN_TYPE_TIME_HHMMSS_24	22
WIN_TYPE_TIME_HHMM_12	23
WIN_TYPE_TIME_HHMMSS_12	24
WIN_TYPE_TIME_DDMMYY	25
WIN_TYPE_TIME_DDMMYYYY	26

3) GIF Display

Upload the GIF using the UI or the application.

Animated GIF.

TTMMFFEESPSTAC

[01U6000000000000/dog2.gifXX]

[01W01,0,0,10,64,32XX]

#Scrolling SingleFrame GIF

Effect EE = SCROLL_ANIM_SCROLL_LEFT

Effect SP = 00 - Speed

Effect ST = 05 - Five Pixel.

TTMMFFEESPSTAC

[01U60000002000500/rstat32.gifXX]

[01W01,0,0,0,128,32XX]

3) Clock with Format.

```
WIN_TYPE_TIME_FORMAT = 27
// TMMFFEESPSTAC
    @ -/+/. Time Offset
    HHMM
    FormatString
[01U27010303010F01+0530H:i:sXX] - With Offset
// One Window 64x16 with Template 1.
[01W01,1,0,0,64,16XX]
```

Other Options

```
// TMMFFEESPSTAC
[01U27010303010F01.a A XX]
[01U27010303010F01.b B XX]
[01U27010303010F01.d e Y yXX]
[01U27010303010F01.d/m/YXX]
[01U27010303010F01.h:i:sXX]
[01U27010303010F01.H:i:s PXX]
[01U27010303010F01.H:i:sXX]
```

Date Time Format Characters Reference.

'H'	hour	14
'h'	hr12	02
'i'	min	03
's'	sec	04
'd'	date	02
'e'	date	2 Without leading 0
'm'	month	03
'n'	month	3 Without leading 0
'Y'	4 digit year	
'y'	2 digit year	
'P'	PM/AM	
'p'	pm/am	
'a'	WeekNamesShort	
'A'	WeekNamesLong	
'b'	MonthNamesShort	
'B'	MonthNamesLong	

4) Display Temperature and Humidity.

```

WIN_TYPE_TEMPERATURE_C 02
WIN_TYPE_TEMPERATURE_F 08
WIN_TYPE_HUMIDITY      03

// SET Temperature Degree C
// TMMFFESPSTAC
// CPSTZA-0+
[01U02010101010F01-02sf00111-XX] //TEMP_C
[01W01,1,0,0,64,16XX]

```

C	Digit Count	0-F(15)
P	Decimal Position	0-F(15)
S	Signed / Unsigned	s => Signed u => Unsigned i => Inverted Hex for Modbus 32 bit
T	DataType	b => byte c => char e => 2char ascii (16bits) s => uint16(short) i =>uint32 f => float(16bit Half - precision) d =>float(32bit)
Z	Leading Zero	1/0
A	Scroll Align	0-4
-	Negative Color	0-F
0	Zero Color	0-F
+	Positive Color	0-F

Temperature Adjustment

```
// SET Temperature Degree C
// TMMFFEESPSTAC
// CPSTZA-0+ W0W1 D1|D2|R1|R2|
[01U02010101010F01-02sf00111.0000|0|1000|0|100|XX] //TEMP_C
[01W01,1,0,0,64,16XX]
```

To set the adjustment values use the W1 to 00.

D1 D2 is the Mapping Range of the Display Data.

R1 R2 is the Range of the Sensor/Register Data.

So if D1-D2 is 0-1000 and R1-R2 is 0-100

If the Sensor generates 50 the value that will be displayed is 500

You can also set the Display Adjustments using the Internal Config Variables which can be set using IR Remote.

```
[01U02010101010F01-02sf00111.0001XX] //TEMP_C
[01W01,1,0,0,64,16XX]
```

Enter the Remote Menu by pressing Menu Button.

Menu ->

RVars ->

DStart -> 0

DEnd -> 1000

RMin -> 0

RMax -> 100

...

Save .

INDUSTRIAL APPLICATIONS (Needs FlinkDAM Hardware Module).

GPIO

GPIO Data Display Boards

2 Windows Showing GPIO Counters

SET GPIOCounter

-GPIO Addr in Decimal

xGPIO Addr in hex

TTMMFFEESPSTAC CPSTZA-0+ RRRR

[01U04040103010F01-30si00111-0000XX]

[01U04050103010F01-30si00111-0001XX]

[01W02,4,0,0,64,16,5,0,16,64,16,XX]

GPIO HILOW MESSAGE

Set Text Message 1

TTMMFFEESPSTAC

[01U010103032F0F01-1GPIOMSG1-XX]

Set Text Message 2

TTMMFFEESPSTAC

[01U010203032F0F01-2GPIOMSG2-XX]

TTMMFFEESPSTAC

CPSTZA-0+

GP GPIO to use

M1 Message ID to play when GP is LOW

M2 Message ID to play when GP is HIGH

[01U05090103010F01-30si00111-060102XX]

if GPIO 6 is LOW then Show Message 01 / HIGH -> Message 02

[01W01,9,0,0,64,32,XX] - Set Windows

GPIO_BIN_MESSAGELOOKUP

```
// SET Multi GPIO Bits combined as a Binary Number
// RR - StartGPIO / RR - Count
```

Setup Messages for GPIO 0..7

```
// TTMMFFEEPSTAC
[01U01000303010F01-GPIO0XX]
[01U01010303010F01-GPIO1XX]
[01U01020303010F01-GPIO2XX]
[01U01030303010F01-GPIO3XX]
[01U01040303010F01-GPIO4XX]
[01U01050303010F01-GPIO5XX]
[01U01060303010F01-GPIO6XX]
[01U01070303010F01-GPIO7XX]
```

-GPIO Addr in Decimal

xGPIO Addr in hex

TTMMFFEEPSTAC-CPSTZA-0+ RRRR

```
[01U06090103010F01-30si00111x0000XX]
```

```
[01W01,9,0,0,32,16,XX]
```

GPIO (Active Low)

0123 Displayed Message

0000 - GPIO0

0001 - GPIO1

0010 - GPIO2

0011 - GPIO3

0100 - GPIO4

0101 - GPIO5

0110 - GPIO6

0111 - GPIO7

.....

GPIO_MESSAGELOOKUP

EACH GPIO HAS A SEPERATE MESSAGE ONLY ONE GPIO AT A TIME

```
// TTMMFFEESPSTAC
[01U01000303010F01-GPI00XX]
[01U01010303010F01-GPI01XX]
[01U01020303010F01-GPI02XX]
[01U01030303010F01-GPI03XX]
[01U01040303010F01-GPI04XX]
[01U01050303010F01-GPI05XX]
[01U01060303010F01-GPI06XX]
[01U01070303010F01-GPI07XX]
```

-GPIO Addr in Decimal

xGPIO Addr in hex

TTMMFFEESPSTAC-CPSTZA-0+ RRRR

```
[01U07090103010F01-30si00111x0000XX]
```

```
[01W01,9,0,0,32,16,XX]
```

```
-----
GPIO   (Active Low)
0123   Displayed Message
-----
```

```
0000 - GPIO0
0001 - GPIO1
0010 - GPIO2
0100 - GPIO3
1000 - GPIO4
```

MODBUS

Display Modbus Values to Window

RRRR Modbus Register Index.

U Modbus Register - signed/unsigned

T Modbus Register - **b** - byte(8bit)

c - char(8bit)

s - short(16bit)

e - 2 ASCII(16bit)
i - int(32bit)
f - float(16bit)-½ Prec
d - double(32bit)

-GPIO Addr in Decimal

xGPIO Addr in hex

TTMMFFEESPSTAC-CPSTZA-0+ RRRRUT

[01U11040103010F01-42sf00111x0000ssXX]

[01ZD01010001XX] - Set Modbus Mode - Slave

[01ZR01030104XX] - Set MODBUS ID ,FUNC ,START ,COUNT => This is only for master.

TTMMFFEESPSTAC-CPSTZA-0+xRRRRUT

[01U11040103010F01-42sf00111x0000ssXX]

[01W01,4,0,0,64,16,XX] - Set 1 Window

[01F9832XX]

SET R0 in Simulator to 16384

SET R1 in Simulator to 17562

/*****

Modbus Raw Data to test in simulator

This will be the data from master to write into slave reg.

// Value =1

\x01\x06\x00\x00\x00\x01\x48\x0A

// Value =4460 => 0x1234

\x01\x06\x00\x00\x12\x34\x84\xBD

*/

#####

2 Windows Showing 2 Registers

#####:ga

[01ZR01030104XX] - Set MODBUS ID ,FUNC ,START ,COUNT

TTMMFFEESPSTAC-CPSTZA-0+ RRRRUT

[01U11040103010F01-32sf00111x0000sdXX]

[01U11050103010F01-32sf00111x0001sdXX]

[01W02,4,0,0,64,16,5,0,16,64,16,XX]

- Set 2 Window

MODBUS_MSGLOOKUP

Use the value in the MODBUS register to lookup a message.

[01ZR01030104XX] - Set MODBUS ID ,FUNC ,START ,COUNT

Set the Messages to be used as Lookup

[01U01000303010F01-MOBUS0-XX]

[01U01010303010F01-MOBUS1-XX]

[01U01020303010F01-MOBUS2-XX]

[01U01030303010F01-MOBUS3-XX]

[01U01040303010F01-MOBUS4-XX]

[01U01050303010F01-MOBUS5-XX]

[01U01060303010F01-MOBUS6-XX]

[01U01070303010F01-MOBUS7-XX]

RRRR Modbus Register Index.

U Modbus Register - signed/unsigned

T Modbus Register - **b** - byte(8bit)

c - char(8bit)

s - short(16bit)

e - 2 ASCII(16bit)

i - int(32bit)

f - float(16bit)-½ Prec

d - double(32bit)

-GPIO Addr in Decimal

xGPIO Addr in hex

TTMMFFEEPSTAC-CPSTZA-0+ RRRRUT

[01U12090103010F01-42si00111x0000usXX]

[01W01,9,0,0,64,16,XX]

- Set 1 Window

MODBUS_TRANSFORM

Display 4-20ma Modbus Analog Current Loop Values with inline mapping.

Ex: 4-20 ma Modbus Transformed Value

[01ZR01030104XX] - Set MODBUS ID ,FUNC ,START ,COUNT

TTMMFFEESPSTAC-CPSTZA-0+ RRRRUT D1|D2|R1|R2|
[01U14040103010F01-32sf00111x0000us|1000.00|2000.00|4000|20000XX]
[01W01,4,0,0,64,16,XX]

D1 D2 is the Mapping Range of the Display Data.

R1 R2 is the Range of the Register Data.

MODBUS_TRANSFORM_RV

Display 4-20ma Modbus Analog Current Loop Values with mapping from UserSettings , set using IR Remote.

Ex: 4-20 ma Modbus Transformed Value from Remote Vars

[01ZR01030104XX] - Set MODBUS ID ,FUNC ,START ,COUNT

TTMMFFEESPSTAC-CPSTZA-0+ RRRRUT
[01U14040103010F01-32sf00111x0000usXX]
[01W01,4,0,0,64,16,XX]

REMOTE_ENTRY Window

This is useful for creating windows that accept User Input for example in production boards for PLAN Quantity. The actual counter comes from the GPIO and the GAP and Efficiency are Calculated Windows.

Use Menu key to Cycle Focus of the Windows.

Enter data with 0-9 keys.

```
[01W02,0,0,0,32,16,1,32,0,32,16,XX]
```

```
// TMMFFEESPSTAC....
```

```
[01U300001033002011234XX]
```

```
[01U300101033002015678XX]
```

```
[01W04,0,0,0,32,8,1,32,0,32,8,2,0,8,32,8,3,32,8,32,8,XX]
```

```
[01U300001013002011111XX]
```

```
[01U30010101300201222XX]
```

```
[01U30020101300201333XX]
```

```
[01U30030101300201444XX]
```

Calculated Windows

```
#####
```

```
WIN_TYPE_SUM // Add Windows Specified
```

```
WIN_TYPE_DIFF // Subtract Windows Specified
```

```
WIN_TYPE_MUL // Multiply Windows Specified
```

```
WIN_TYPE_DIV // Divide Windows Specified
```

```
WIN_TYPE_EFF //  $EFF=(W0/W1)*100.0;$ 
```

```
WIN_TYPE_TRANSFORM // Similar to Modbus Transform for W0
```

```
#####
```

```
[01W04,0,0,0,32,8,1,32,0,32,8,2,0,8,32,8,3,32,8,32,8,XX]
```

```
[01U04000103010F01-30si00111-0000XX] - Define GPIO Counter
```

```
[01U31010101300201-50si00111-1000XX] - Define the Target IR-Remote Data Entry  
Window - Def Value 100
```

```
// TMMFFEESPSTAC CPSTZA-0+ WAWB - WA and WB are the Input Windows.
```

```
[01U41020101300201-30si00111-0100XX] - DIFF => D1-D0 - Target-Count
```

```
[01U44030101300201-22sd00111-0001XX] - EFF =>  $(W0/W1)*100$ 
```

```
EFF = (Count/Target)*100
```

Using Remote Vars for Calculations.

To use the Remote Variables for the Calculation you can set WA and WB to 90-98
When WA and WB are 90-98 then the Remote Vars are used for calculation.

90 => "DStart"
91 => "DSEnd"
92 => "RMin"
93 => "RMax"
94 => "T1"
95 => "T2"
96 => "T3"
97 => "T4"
98 => "V8"
99 => "V9"

Set T1 to 100 and test the following.

```
[01W02,0,0,0,32,8,1,32,0,32,8,XX]  
// TTMMFFEEPSTAC CPSTZA-0+ WAWB - WA and WB are the Input Windows.  
[01U04000103010F01-30si00111-0000XX] - Define GPIO Counter  
[01U40010101300201-30si00111-0094XX] - SUM => W0+T1
```

WIN_TYPE_TRANSFORM

```
WAWB  
[01U04000103010F01-30si00111-0000XX]  
- Define GPIO Counter in Window 0  
[01U45010101300201-22sd00111-0000|100|200|0|50|XX]  
- Generate TRANSFORMED data from WA(Window0), Mapping Data in the packet  
  
[01W02,0,0,0,64,8,1,0,8,64,8,XX] - Display  
Calculated Value in => W1 (Result)
```

WIN_TYPE_TRANSFORM_RV

Uses the Transformation data from Remote Vars.

Goto Settings by Pressing Menu KEY

Press Up/Down till you get RVars. Press Enter

Set

```
DMin=100    DMax=1000    (Display)
RMin=1      RMax=100    (Register Value)
```

These values will be used for calculating the value in the target window

```
[01U04000103010F01-30si00111-0000XX]    - Define GPIO Counter in Window 0
//    TMMFFEESPSTAC CPSTZA-0+ WAWB        - WA= Source Window from where
the data is taken
[01U46010101300201-22sd00111-0000XX]    - Generate TRANSFORMED data from
WA(Window0) , Mapping from RVars
[01W02,0,0,0,64,8,1,0,8,64,8,XX]        - Display Calculated Value in => W1
(Result)
```

Timer Boards

Goto Settings by Pressing Menu KEY

Press Up/Down till you get T1. Press Enter

Here set the TIMER_DURATION.

For Count Down TIMER_DURATION is where the timer starts and counts down to 0

For Count UP TIMER_DURATION is where the timer Ends counting up from 0.

T2 is where you set the Buzzer Duration for the Buzzer when the Timer reaches the end.

Leave this to 0 for no Buzzer.

T3 and T4 are the start and stop time for Buzzer when the timer is running.

In some cases you will want the buzzer to come on at say 60-70 Seconds and the counter to continue counting. For this you can set TIMER_DURATION to a large number and

T3=60 and T4=70.

This can also be for like warning. Where TIMER_DURATION=120 and T3=100 T4=100
So 20 Seconds before the timer runs out there is a warning Buzzer.

```
//   TTMMFF.....CUL...
//           *   Timer Type U=> CountUp Timer D=> Countdown Timer.
//           *   Latch Timer.
//   TTMMFFEESPSTAC
//           U   Timer Type U=> CountUp D=> Countdown.
//           L   Timer Type .=>Pulse L=> Latch
[01U50000101300201ULXX]
```

Ex: Pulse Count Up Timer

```
[01W01,0,0,0,32,16XX]
//   TTMMFFEESPSTACUL
[01U50000101300201U.XX]
```

Pulse Count Down Timer

```
[01W01,0,0,0,32,16XX]
//   TTMMFFEESPSTACUL
[01U50000101300201D.XX]
```

Latch Countdown Timer

```
[01W01,0,0,0,32,16XX]
//   TTMMFFEESPSTACUL
[01U50000101300201DLXX]
```

Latch CountUP Timer

[01W01,0,0,0,32,16XX]

TTMMFFEESPSTACUL

[01U50000101300201ULXX]

[01F9832XX] - Commit.

SPECIAL COMMANDS
#####

Set Time

HHMMSS

[01S213600XX]

SET DATE

DDMMYY

[01D010722XX]

Dump Current Time

[01ZTXX]

Save Window and Message data to Flash with PIN-9832

[IDF9832CRC]

[01F9832XX]

Set MsgDisplayDuration

[ID ZB MsgDisplayDuration CRC]

[01ZB05XX]

Set Brightness

[ID ZH Brightness CRC]

[01ZH1XX]

Set Idle Interval

[ID ZA Idle Interval CRC]

[01ZA05XX]

Dump Board Info

[ID ZU CRC]

[01ZUXX]

Dump Window Information

[ID ZW S WW CRC]

S=0 => AllWindows , S=1 => One Window

[01ZW000XX]

[01ZW101XX]

[01ZW115XX]

Dump Message Information

[ID ZM WW CRC]

[01ZM00XX]

[01ZM01XX]

[01ZM15XX]

SET MODULE WIDTHxHEIGHT

T - Module type 0-32x16 1-32x32 2-64x32

W - Module Width (Number of Modules Horizontally)

H - Module Height (Number of Modules Vertically)

[01ZZ01100000XX]

[01ZZ01100000XX] [01F9832XX] [01ZX9732XX]

```
# HUB 75 64x32 2x2 => 128x64
[01ZZ22200000XX] [01F9832XX] [01ZX9732XX]
[01ZUXX]
```

```
# HUB 75 64x32 1x1 => 64x32
[01ZZ21100000XX] [01F9832XX] [01ZX9732XX]
[01ZUXX]
```

```
// 2x 1 module
[01ZZ22100000XX] [01F9832XX] [01ZX9732XX]
```

```
#####
```

Set Modbus RegCount

[ID ZR ID FN ST CT CRC]

 ID - Modbus 01

 FN - Function 03

 ST - Start 01

 CT - Reg Count 04

[01ZR01030104XX]

Set the Device ID BaudRate PollRate ModbusMode

[01ZD ID BD PL CM XX]

ID - Device ID -01

BD

00: 115200

01: 1200

03: 4800

04: 9600

05: 19200

06: 38400

07: 57600

C - Clock Mode : 0-NONE 1-Slave 2-MASTER

M - Modbus Mode : 0-NONE 1-Slave 2-MASTER

Set the Device ID, Baud Rate etc.

 ID - Device ID

BD - Modbus BaudRate
PL - Modbus Pool Interval
C - CLOCKMODE 0-NONE 1-Slave 2-MASTER
M - Modbus Mode 0-NONE 1-Slave 2-MASTER

[01ZD01010001XX]

RESTART BOARD

[ID ZX 9732 CRC]

[01ZX9732XX]

Special Commands that will work only from serial port

Set Wifi Access Point

[ID ZS SSID CRC]

[01ZSMyAccessPointXX] [01ZX9732XX]

Set Wifi PWD

[ID ZP PWD CRC]

[01ZP1234xyzXX]

This sets the IP TYPE TO DHCP

Reset SSID

[01ZSXX] [01ZX9732XX]

Set Wifi IPADDR

[ID ZI IPADDR CRC]

[01ZI192.168.3.39XX]

This sets the IP TYPE TO STATIC and netmask to 255.255.255.0

Set Palette Color Commands.

[IDZC

A - (A)ssign Color
(S)ave Palette
(D)ump.
(C)lear and Restore default - Restart after this command.

I - Color Index 0-F

XXXX - 565 Color in Hex Format 1234

CRC]

[01ZCA1857DXX]

[01ZCSXX]

[01ZCDXX]

[01ZCCXX]

Program Commands.

You can add protocol commands into a file named p_0000.txt .. p_0999.txt
And Set Auto Program to 1 and the board will start playing the programs This
can be done from UI or from the commands.

Next Program Delay Command.

[IDN

dddd - Next Program Load Delay in seconds.

CRC]

[01N0003XX]

Last Program Delay Command.

[IDE

dddd - Last Program Load Delay in seconds.

CRC]

[01E0003XX]

This command allows setting of the delay for each program. This should be the last command in the program file. When playing from programs, the system does not save the window (V Command) or messages (U Command). When delay command E is specified its assumed that that is the last program and the next program will start from P_0000.txt.

Example Programs

P_0000.txt

```
[01W01,0,0,64,32,XX]  
[01U600000000000000/gan.gifXX]  
[01N0003XX]
```

P_0001.txt

```
[01W01,0,0,64,32,XX]  
[01U600000000000000/krish.gifXX]  
[01N0003XX]
```

P_0002.txt

```
[01W01,0,0,64,32,XX]  
[01U600000000000000/krish1.gifXX]  
[01E0003XX]
```

Set IsAutoRunProgram start / stop.

```
[IDZG  
    V -    1 - Start AutoProgram / 0 - Stop AutoProgram  
    CRC]  
[01ZG0XX]  
[01ZG1XX]
```

Play One Program Command.

This command will play a stored program file.

p_0004.txt

```
[01W02,0,0,32,8,0,8,32,8,XX]  
[01T01000103400201ABCDEFGHJKLMNOPQRSTXX]  
[01T01010103400201ABCDEFGHJKLMNOPQRSTXX]  
[01E0003XX]
```

Play Program .

```
[IDA  
    dddd - Program to play.  
    CRC]  
[01A0004XX]
```

Command Reference.

Command	Description
[01R I XX]	Remote Commands.
[01S 191000 XX]	Set Time .HHMMSS
[01D 010722 XX]	Set date DDMMYY
[01A 0001 XX]	Play One Program.
[01F 9832 XX]	Save Changes
TTMMFFEESPSTAC [01U 01000103300201 -0Default-XX]	Save Message.
TTWWFFEESPSTAC [01T 01000104300202 ABCDEFSTXX]	Set TEXT to Window.
[01W 01,0,0,0,10,10 ,XX]	Setup Window
[01V 01,0,0,0,10,10 ,XX]	Setup Window and Save to Flash WindowCount,x1,y1,w1,h1,x2,y3,w3,h4,...
MMWWC [01L 00011 XX]	LoadMessageToWindow [ID L Message(0-64) Window(0-15) C=>Override Color - used if Color>0 CRC]
WWC [01P 001 abcdefghijXX]	Replace TEXT [ID P Window(0-15) C=>Color tttttt CRC]
[IDN dddd XX]	Next Program Delay. This will start next after the delay.
[IDE dddd XX]	Last Program Program Delay. This will start program 0 after the delay.
[01ZA 05 XX]	[ID ZA idle Interval CRC]
[01ZB 05 XX]	[ID ZB MsgDisplayDuration CRC]

[01ZCA11234XX]	[IDZC A - Assign Color ; S - Save Palette. ; D - Dump Palette. C - Clear Palette. C - Color Index 0-F XXXX - 565 Color in Hex XX CRC] [01ZCA11234XX]
[01ZCSXX]	SavePalette
[01ZCDXX]	DumpPalette
[01ZCCXX]	ClearToDefaultPalette
[01ZDIBDPLSLXX]	Device ID and Modbus Params. ID - Device ID -01 BaudRate 00: 115200 01: 1200 03: 4800 04: 9600 05: 19200 06: 38400 07: 57600 CLOCKMODE 0-NONE 1-Slave 2-MASTER Set the Device ID ID - Device ID BD - Modbus BaudRate PL - Modbus Pool Interval C - CLOCKMODE S - Modbus IsSlave [01ZD01010021XX]

[01ZFLXX]	Dump File List
[01ZFD/result.gifXX]	Delete File
[01ZFR/ffffXX]	Read File
[01ZFWFF1234 binary data of SIZE XX]	<p>Write To File</p> <p>SIZE - Size in Hex</p> <p>FNUM- File Number (hex)</p> <p>data</p> <p>[01ZFW0005ffff12345XX]</p> <p>[01ZFW0005ffff12345690XX]</p> <p>[01ZFW100Affff12\[\]3456789.XX]</p> <p>[01ZFW100Affff12\[\]3456789.XX]</p> <p>if (flen&0xF000) isAppend=1;</p>
[01ZG0XX] [01ZG0XX]	<p>Set IsAutoRunProgram start / stop</p> <p>// [01ZG0XX]</p> <p>// [01ZG1XX]</p>
[01ZH1XX] [01ZHFX]	<p>Set Brightness</p> <p>Brightness - 0-F</p> <p>[ID ZH B CRC]</p>
[01ZIXX]	Clear All Windows
[01ZM 00 XX]	<p>Dump Message Info</p> <p>[ID ZM MM CRC]</p>
[01ZR 01020104 XX]	<p>// Set Modbus RegCount</p> <p>[ID ZR ID FN ST CT CRC]</p> <p>ID - Modbus 01</p> <p>FN - Function 03</p> <p>ST - Start 01</p> <p>RC - Reg Count 04</p> <p>[01ZR01020104XX]</p>

[01ZTXX]	Dump Time
[01ZUXX]	Dump All
[01ZW100XX]	Dump Window Info [ID ZW S WW CRC] S=1 => Short or S=0 => Full WW [01ZW100XX]
[01ZX9732XX]	Restart Board.
[01ZSwifissidXX]	Set Wifi SSID
[01ZPwifipinXX]	WiFi PIN
[01ZI192.168.3.88XX]	Wifi IP