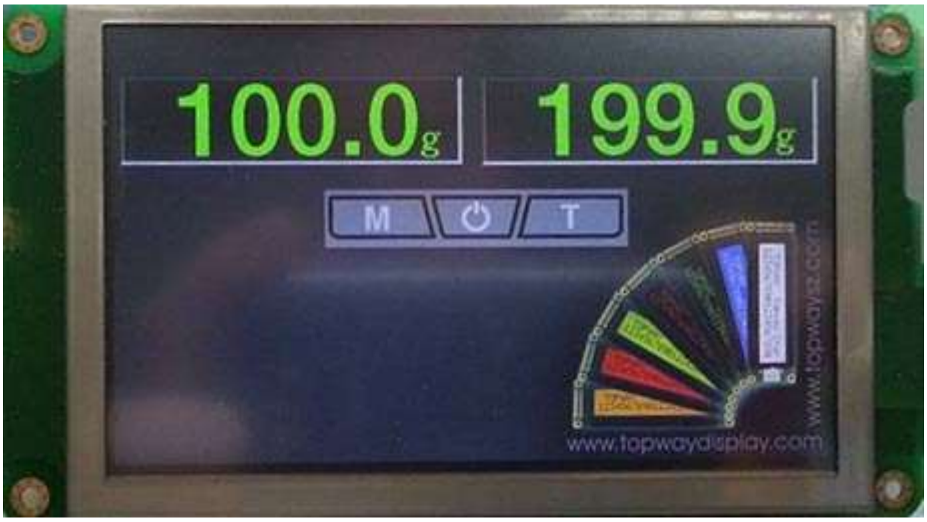
Title: Incorporating HMT050ATA-2C as a Digital Scale Display	by: A.S.	Date: 2020-06-12

1. Background

The system we are going to build includes a data acquisition circuit with a host MCU and a HMT050ATA-2C smart LCD module as scale's display.



HMT050ATA-2C is a smart LCD module with following specifications:

- 5 inch TFT Color LCD with 800 (RGB) x 480 pixels with 16 bits color per pixel and LED backlight
- High-resolution resistive touch panel for user inputs and interactions
- 32 bits ARM Micro-Controller with a Smart LCD Engine to process graphics, commands from host
- A large flash memory to hold graphical data and configuration information locally
- A RS232 or a Logic-Level Serial Interface, connect the module to any host MCU or any other device supporting serial communication
- A USB interface to configure the module

The HMT050ATA-2C provides a very easy and fast way to create high-end embedded applications for Industrial Controls, Instruments, and Medical devices. The TFT LCD module shows user interface, which can be created with Topway's UI development tool. A host MCU may connect to the module, control it and receive input events.

Information showed is for guidance only; never use as a reference design.
TOPWAY will not responsible for any damage or loss on using the provided information. Please evaluate before using.

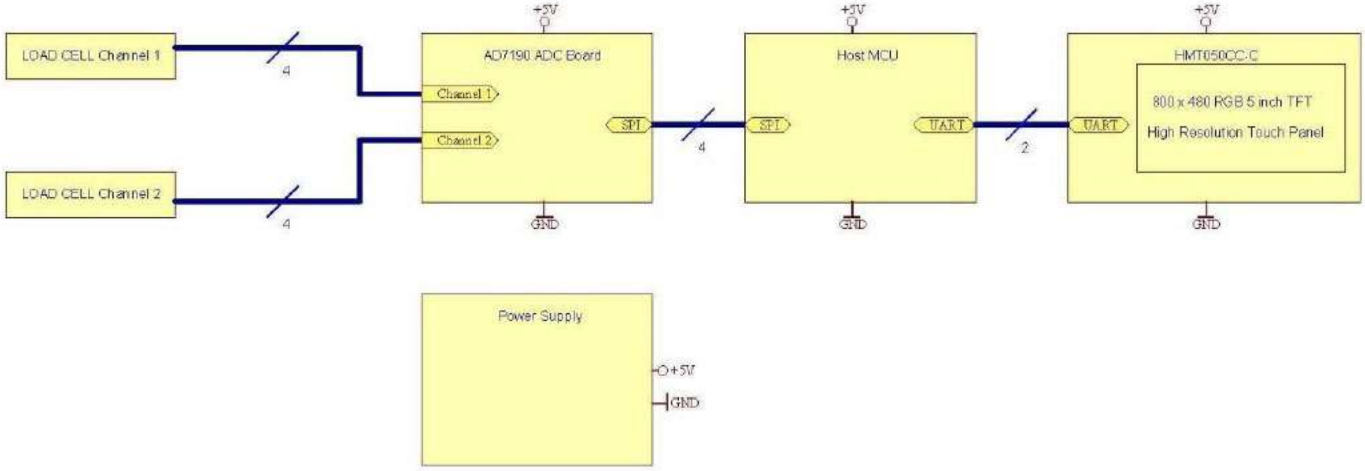
2. Content

2.1 System Overview

A Digital scale is a very good example explaining the functionality and the way of developing applications with HMT050ATA-2C.

The proposed Digital Scale shall include:

- Two Load Cells
- ADC Board (e.g. based on AD7190) with SPI
- Host MCU Board (e.g. based on a small ARM Cortex M0, PIC or AVR) with SPI and UART
- Smart LCD Module HMT050ATA-2C with UART
- Power Supply



The ADC Board is providing following functionality:

- Converting the bridge voltages of two load cell channels
- A ratio-metric measurement circuit is recommended

The Host MCU is providing following functionality:

- Setup and sampling of ADC values from the ADC Board
- Calculating the actual weights from the ADC values
- Transmitting the weight values as 32 bits signed values to the Smart LCD
- Checking some variables inside the Smart LCD and adjusting Types of operation accordingly
- Recording Tare values in Tare Mode

The HMT050ATA-2C is providing following functionality

- Displaying different pages of contents depends on the operation type of the system
- Displaying weight values calculated inside the Host MCU
- Setting internal variables depends on Touch Panel events

The Power Supply is providing following functionality

- Supply a single stable 5V supply of 750mA … 1A depends on the actual circuitry. The Supply Voltage may range from 4.8V … 5.2V DC.

The Host MCU may communicate with the Data Acquisition board via SPI or any interface if required.
The Host MCU also connects to the Smart LCD via a asynchronous serial interface in the following format:

- 8 Bit
- No Parity
- 1 Stop Bit.
- Additional a flow control may be used by setting Jumpers JP3 and JP4. Please refer to the HMT050ATA-2C user manual for more details.
- The Host MCU may use a baud rate of 9600 Bd or 115200 Bd. The baud rate can be selected on the HMT050ATA-2C by using Jumpers JP1 and JP2:
  - JP1 shorted / JP2 open select 115200 Bd
  - JP2 shorted / JP1 open select 9600 Bd

Note: A variety of other Baud Rates may be set via a Serial Communication command. Please refer to the HMT050ATA-2C User Manual for more details.

2.2 The HMT050ATA-2C Smart LCD Module

Smart LCD module has the following parts and functions:

- Contents and functionalities are organized in one (or more) page(s)
- Page is a visual display on screen, contains:
  - Images
  - Icons
  - Variables (VP)
  - Animations
  - Time and Date using internal Real Time Clock
- Page can hold multiple touch sensitive areas, which may trigger events like:
  - Transition to another page
  - Showing specific icons
  - Setting variable value or performing calculation
  - Sending touch event via serial interface
  - Sounding internal buzzer
- Page may initialize variable (VP)
- User interaction or serial interface command can control smart LCD module transition from showing one page to another one
- The module includes a memory space, able to store global variables (VP), such as:
  - Numeric value
  - String
  - Graph
  - Bitmaps
- Through serial interface user can do:
  - Switching between pages
  - Text output
  - Graphic operations like drawing line, rectangle and fill
  - Reading and writing of variable
  - Reading current page id
  - Reading and updating internal real time clock
  - Receiving touch event
  - Setting backlight and buzzer
- Save values to internal flash storage

Development Plan

Define scale's possible operation types and functions
Design a Data Acquisition Board with ADC and SPI interface (e.g. refer to AD7190 datasheet)
Create a simple Host Interface Board with SPI connecting to the data acquisition board, and a serial interface that can be used to communicate with HMT050ATA-2C
Create a design project with multiple pages using Topway UI Editor - RG Tools, which provides the Graphical User Interface

and set internal variables (VP) depends on user interactions
Create communication software for the Host Board, which can sample ADC data, calculate weight and access the internal variables (VP) of HMT050ATA-2C

2.3 The Types of Operation and Functions of a Digital Scale

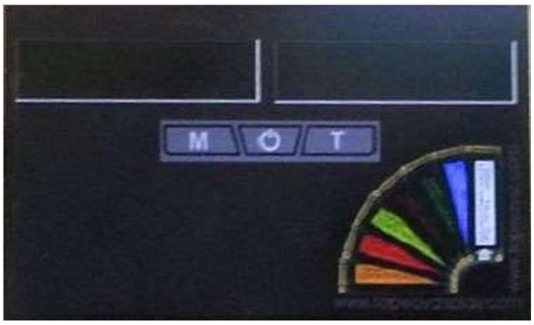
The Host Microcontroller (MCU) is the heart of proposed system. This document does not go into detail on how to implement operational types inside the Host MCU. However, it shall give an overview about all necessary operational types to implement a Digital Scale and how the Host Controller and HMT050ATA-2C may communicate.

Depends on the operational Type, the HMT050ATA-2C displays different contents and Touch Events may trigger certain processing or change in Type.

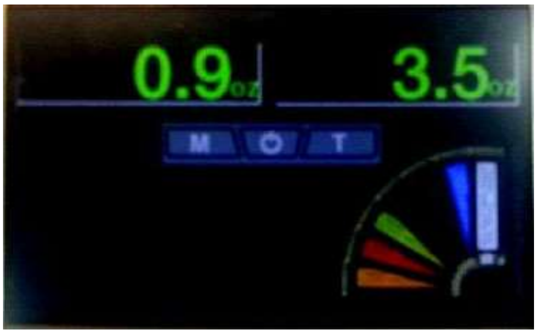
Types of Operation and Functions are:

Off Type (Page 1)

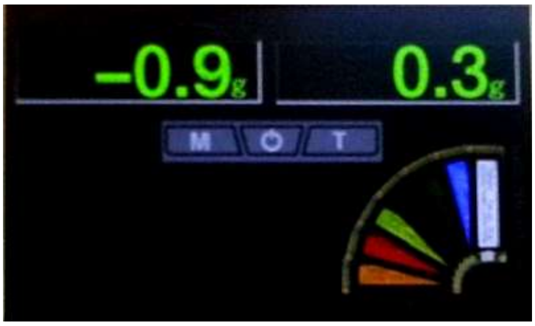
- In Off Type the LCD will show a screen without any weight
- The touch panel On / Off area is active to restart and transit to Normal Type
- The Host MCU shall frequently read an Event register, determine whether the On / Off button is pressed



Normal Type (Page 2)

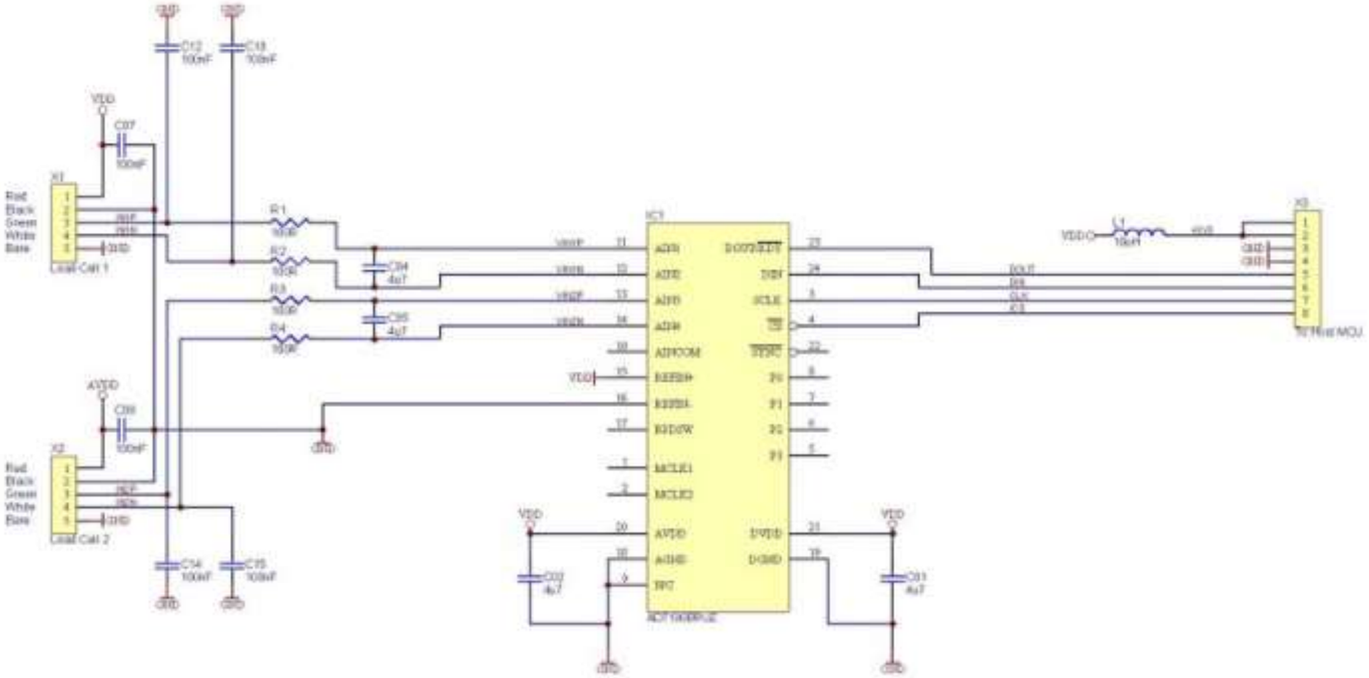


- In Normal Type the Weights on display will be updated every 300ms
- All three Touch Panel key areas (Mode, On / Off, Tare) are active
- Pressing Mode will toggle the unit (g / oz)
- Pressing On / Off will transit to Off Type
- Pressing Tare will Tare the Scale
- Host MCU shall frequently read the Event register to determine whether any of the three Buttons was pressed
- Host MCU samples the load cell data and calculates the weights (taking unit and Tare values in account)
- Host MCU transmits numerical values of the weights to HMT050ATA-2C by storing those values into two of the HMT050ATA-2C 32 bits variables
- The Host MCU also needs to verify that the unit settings (g or oz) correspond with the currently selected unit setting. If not, it has to update the unit, which is stored as a short string inside the HMT050ATA-2C.



2.4 A design proposal of a SPI Data Acquisition Board

The Data Acquisition Board may be based on any suitable ADC with SPI or any other suitable interface. Ratio-metric measurement is preferred. Following is a proposed circuit for reference.



The circuit is based on the two-channel precision ADC AD7190. It can be connected to via SPI.

2.5 Recommendation on Host MCU

Between the ADC and HMT050ATA-2C is a host MCU. This MCU can be any 4, 8-bit or even 32-bit ARM based. It should provide hardware SPI and UART or software emulated SPI and UART. It does not require lots of resources.

This document does not specify the details of this MCU. Designer is free to choose the appropriated chip. A simple 8051 based MCU or a Cortex-M0 may be used. Additional functionality can be added by engineer.

2.6 Step by step description for creating a Digital Scale’s Graphical User Interface with Topway’s UI Editor and how to upload design project to the HMT050ATA-2C

As mentioned earlier, the HMT050ATA-2C will present the Types of Operation with pages. We need to design these pages. Topway’s RG Tools is the tool we use.

A page on Smart LCD Module is a collection of graphical and touch related elements such as:

- Background Images
- Icons
- Text and Numerical Outputs
- Animations
- Graphs
- Touch Areas

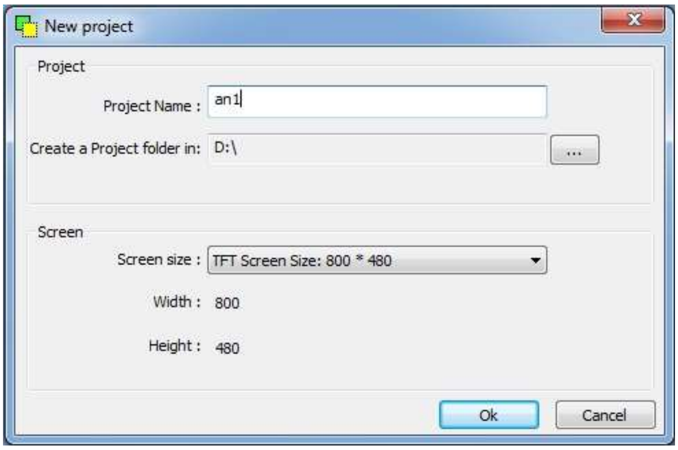
A page usually represents an operation screen.

Besides organizing pages, the UI Editor can create and initialize variables (VP). These variables can be accessed and modified either by events corresponding to the graphical element on a page or by command from serial port (by the Host MCU).

At first, we create a new project by defining:

- Project name, e.g. an1
- Project directory
- Screen Size (800 x 480)

Note: After creating a new project these three Properties are fixed and can’t be modified anymore.



New project

Project

Project Name : an1

Create a Project folder in: D:\

Screen

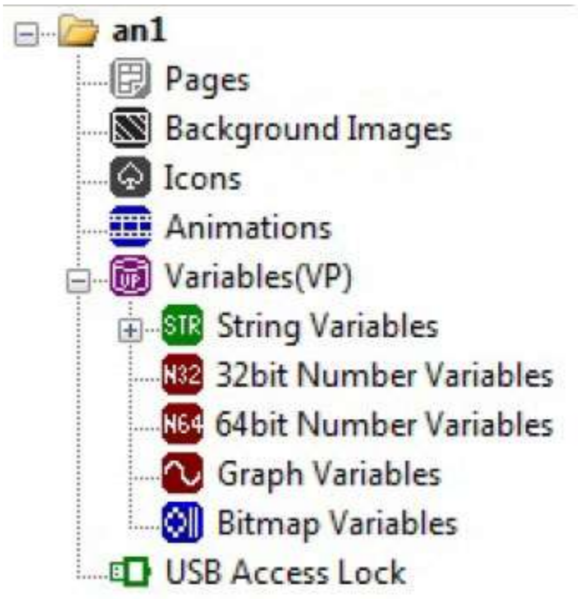
Screen size : TFT Screen Size: 800 * 480

Width : 800

Height : 480

After creation of new project, a list of items becomes visible on the left side of the work space. These are the components of a Smart LCD design project, include pages, resources, and variables:

- Pages manages all pages used in the project
- In Background Images, user can upload and organize all background pictures needed for pages
- Under Icons all used icons can be loaded and organized
- Animations may include animation sequences
- Different kind of variables are allowed:
  - 0-Terminated Strings with up to 128 characters
  - 32-bit signed integer
  - 64-bit signed integer
  - Variables related to Graphs
  - Bitmaps

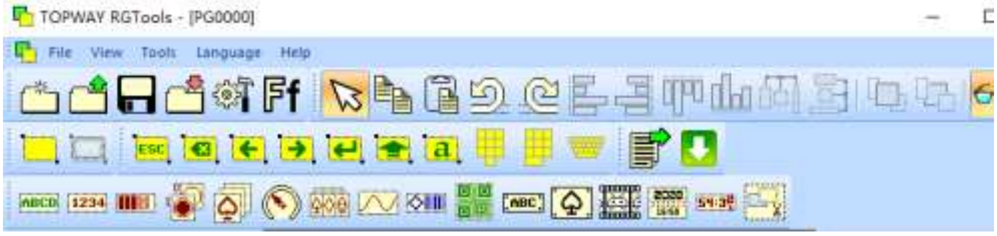


In this project we will use following items:

- Pages
- Background Images
- Icons
- String Variables
- 32-bit signed integer

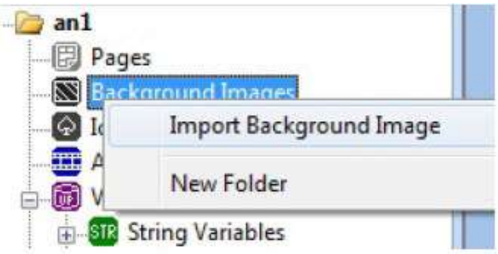
Editor's top icon bar contains varieties of objects. For this project, we will use following features:

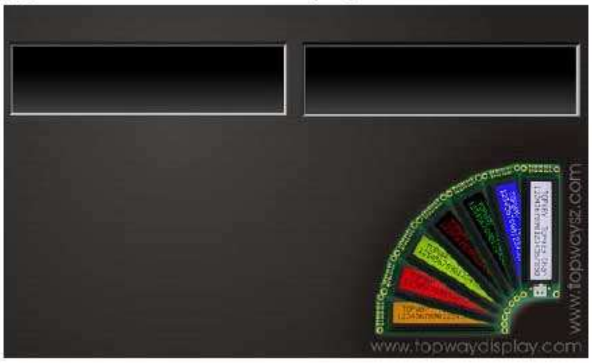
- Select an Object
- Insert Touch Area
- Insert VP String Area
- Insert numerical VP Area
- Compile & Download to Module



After the creation of the project we will define which images and icons we want to use, also we will need to define the variables we need.

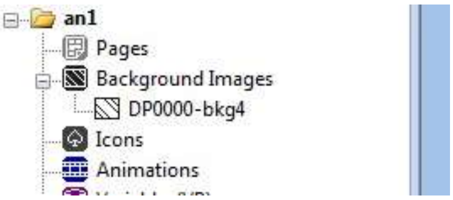
1.) Import Background Images by clicking the Right Mouse Button
Our project will use just one single Background Image (bkg4.jpg), which is used on all pages.



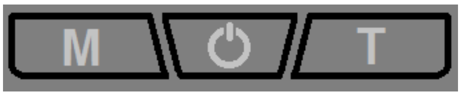


bkg4.jpg

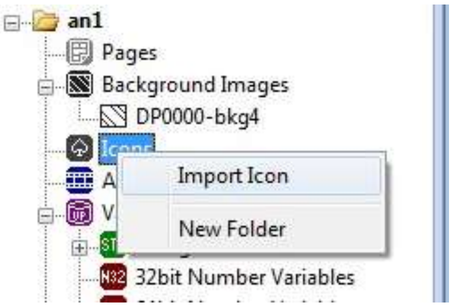
The new Background Image appears as resource in the Background Image list



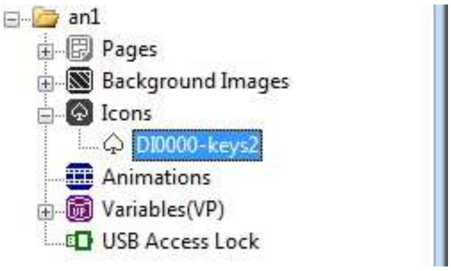
2.) Next step is importing of icons.
We will import just one icon. This icon (keys2.jpg) holds three keys we will need to operate our Digital Scale.



keys2.jpg

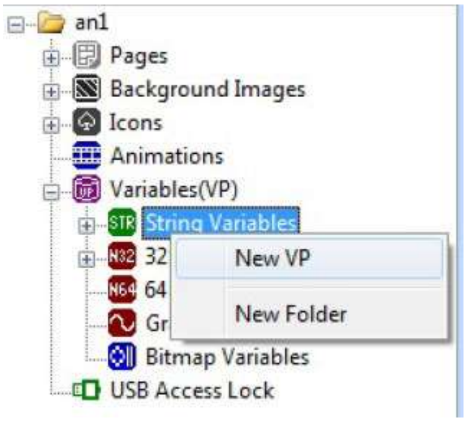


The keys are Mode, On/Off and Tare
Our new icon becomes listed under Icons

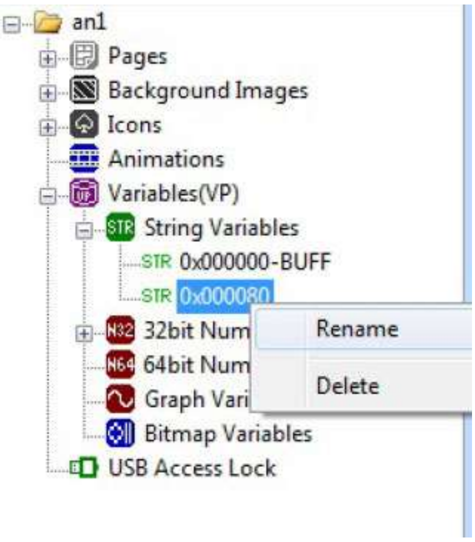


3.) Several different variables (VP) are needed. We create following variables:

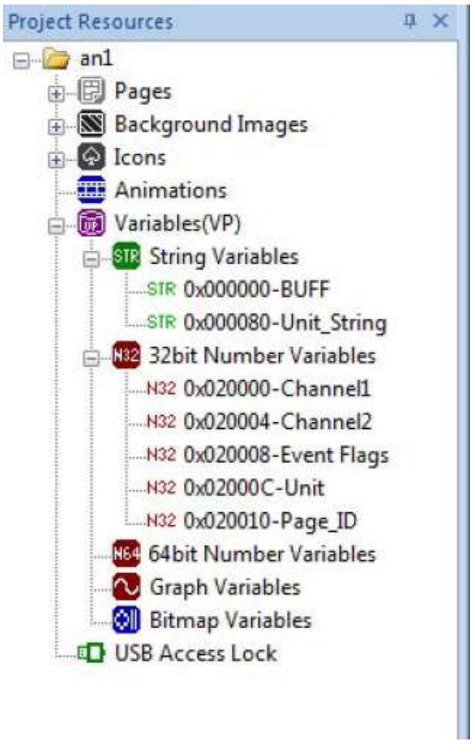
- String Variable Unit_String
- 32-bit Number Variable Channel1
- 32-bit Number Variable Channel2
- 32-bit Number Variable Event Flags
- 32-bit Number Variable Unit
- 32-bit Number Variable Page_ID



After creating a new variable only the variable address is listed without any name. We should assign a name to each variable. Names / alias are very useful distinguishing different variables. Right click on a variable, the Rename function becomes visible and allows user assign or rename a variable Name.



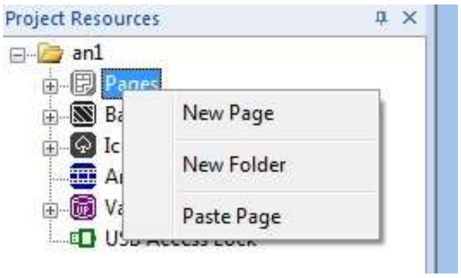
After creating and renaming all variables we should be able to see the following list of variables:


Project Resources
- an1
  - Pages
  - Background Images
  - Icons
  - Animations
  - Variables(VP)
    - String Variables
      - STR 0x000000-BUFF
      - STR 0x000080-Unit_String
    - 32bit Number Variables
      - N32 0x020000-Channel1
      - N32 0x020004-Channel2
      - N32 0x020008-Event Flags
      - N32 0x02000C-Unit
      - N32 0x020010-Page_ID
    - 64bit Number Variables
    - Graph Variables
    - Bitmap Variables
  - USB Access Lock

The variables have following usage:

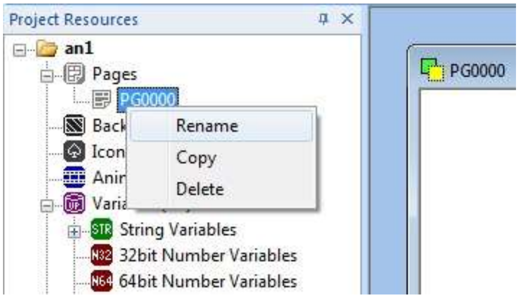
- Unit_String: Contains a string of weight unit shown on LCD, which may be “g” or “oz”
- Channel1: Includes the Weight of Channel 1 (in 0.1g or 0.1 oz steps) shown on the left side of the LCD
- Channel2: Includes the Weight of Channel 2 (in 0.1g or 0.1 oz steps) shown on the left side of the LCD
- Event Flags: Includes Flags representing which Button was pressed. Bit0 = 1 => Tare was pressed, Bit1 = 1 => Mode was pressed, Bit7 = 1 => On/OFF was pressed
- Unit: Includes a numerical representation of the unit (g or oz), with 0 representing oz and any other value g.
- Page_ID: Includes the numerical representation of the current page

4.) After creating of all variables we can work on pages.
The first page represents scale in Off Type.
The second page represents the product while in Normal Type.
Clicking the right mouse button on Pages allows the user to create a new (empty) page.

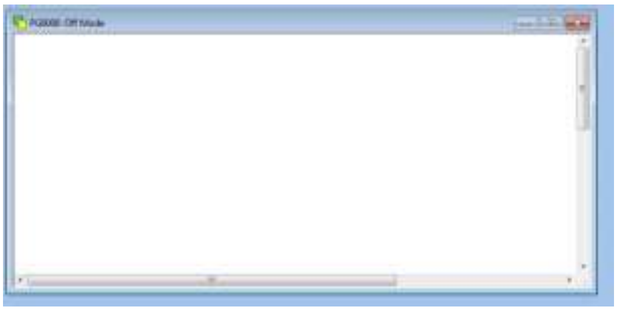


We will create two new pages: PG0000 and PG0001

The first new page is called PG0000. We can rename the page to Off Type:

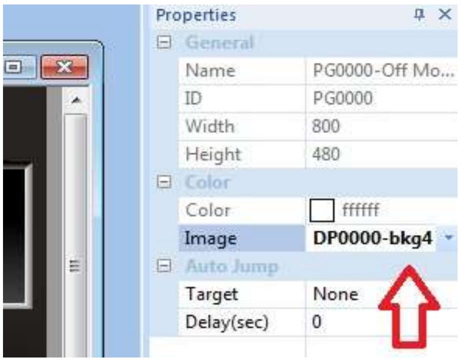


→



PG0001 we will rename to Normal Type.

Now we assign the background image bkg4.jpg to our new page. The background image will immediately be visible on PG0000-Off Type or PG0001-Normal Type.

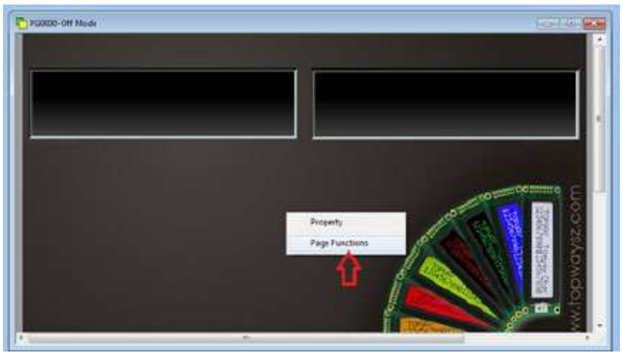

Properties

General	
Name	PG0000-Off Mo...
ID	PG0000
Width	800
Height	480
Color	
Color	□ ffffff
Image	DP0000-bkg4
Auto Jump	
Target	None
Delay(sec)	0

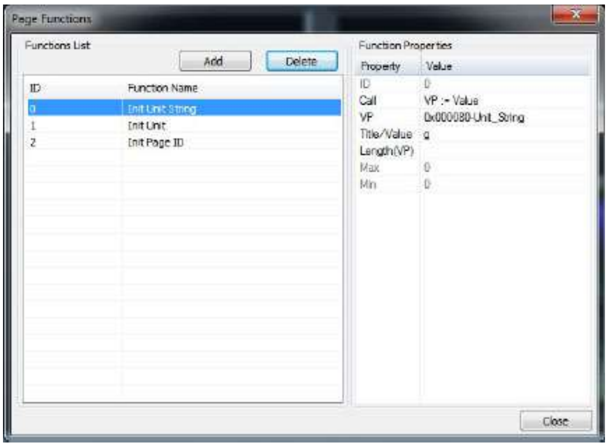
Next we set three initial values for some of our VPs when the page PG0001-Off Type gets called:

- VP Unit_String to “g”
- VP Unit to 0
- VP Page_ID to 0

Right clicking on our page will allow us to call and modify “Page Functions”



We add three different Page Functions On PG0001-Off Type and name them “Init Unit String”, “Init Unit” and “Init Page ID”.
On the right hand side we select the desired functionality.



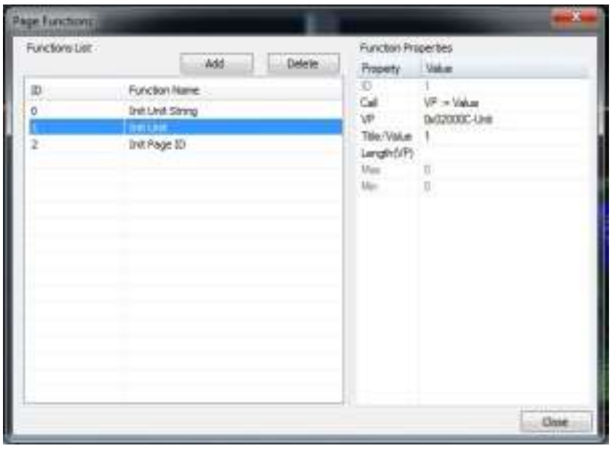
ID	Function Name
0	Init Unit String
1	Init Unit
2	Init Page ID

Property	Value
ID	0
Call	VP := Value
VP	0x000080-Unit_String
Title/Value	g
Length(VP)	
Max	0
Min	0

For “Init Unit String” we choose:

- Call: VP := Value
- VP: 0x000080-Unit_String
- Title/Value: g

We will use “g” to represent gram.
Gram will be the initial unit shown.



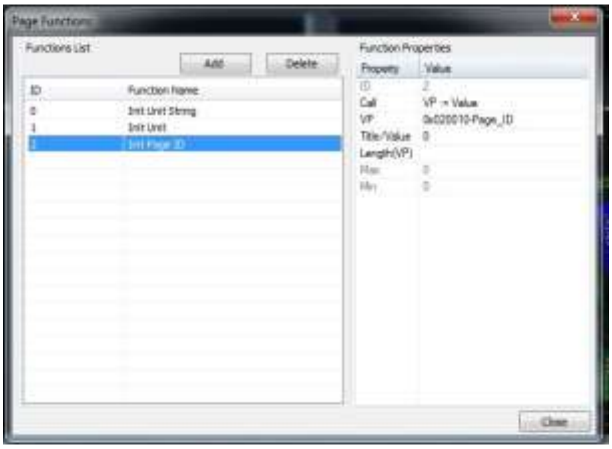
ID	Function Name
0	Init Unit String
1	Init Unit
2	Init Page ID

Property	Value
ID	1
Call	VP := Value
VP	0x02000C-Unit
Title/Value	1
Length(VP)	
Max	0
Min	0

For “Init Unit” we choose:

- Call: VP := Value
- VP: 0x020000C-Unit
- Title/Value: 1

1 represents gram as unit. The Host MCU may read this value to determine what type of data it needs to provide, gram or ounce.



ID	Function Name
0	Init Unit String
1	Init Unit
2	Init Page ID

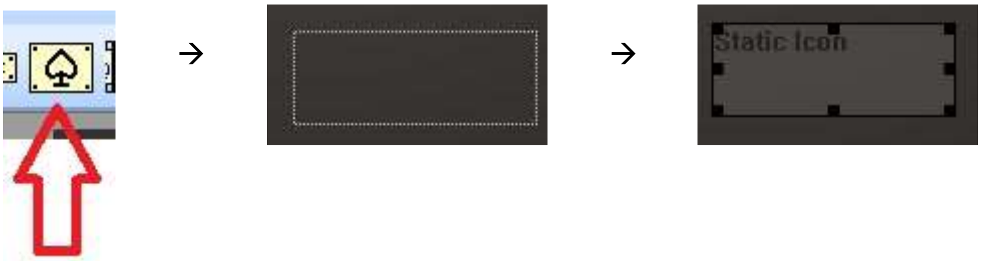
Property	Value
ID	2
Call	VP := Value
VP	0x020010-Page_ID
Title/Value	0
Length(VP)	
Max	0
Min	0

For “Init Page ID” we choose:

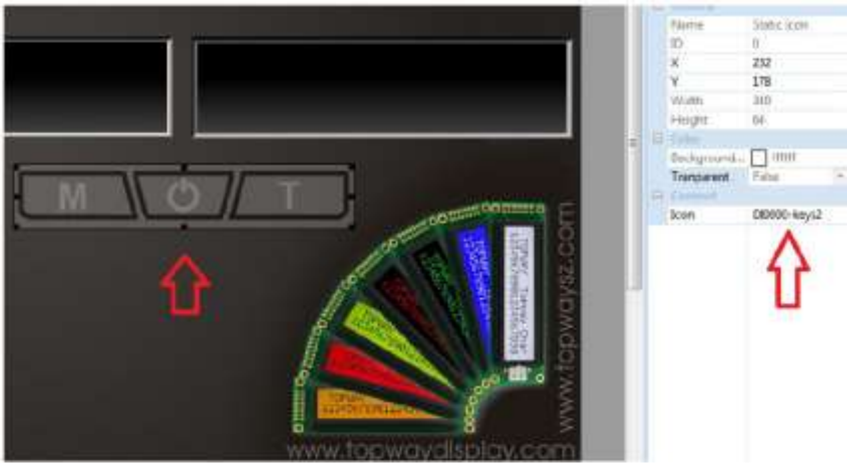
- Call: VP := Value
- VP: 0x0200010-Page_ID
- Title/Value: 0

0 indicates that the current shown page Is page PG0000. The Host MCU may read this value to decide in which type the system is currently in.

Now we are ready to place our first icon by clicking on the Static icon symbol and placing it somewhere onto our page. It acts as a place holder for icon image.



Next, we select the icon and set X to 232, Y to 178 and assign icon-image DI0000-keys2 to it.

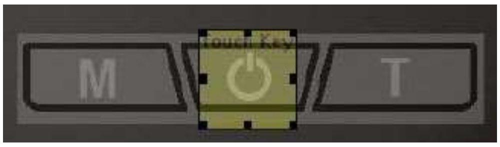


Name	Static Icon
ID	0
X	232
Y	178
Width	310
Height	64
Background...	□
Transparent	False
Icon	DI0000-keys2

The icon image will be automatically placed onto the right spot and is shown on our page.

While in page PG0000-Off Type the system is waiting for a touch event from the ON/OFF button. The Mode and Tare buttons won’t trigger any event.

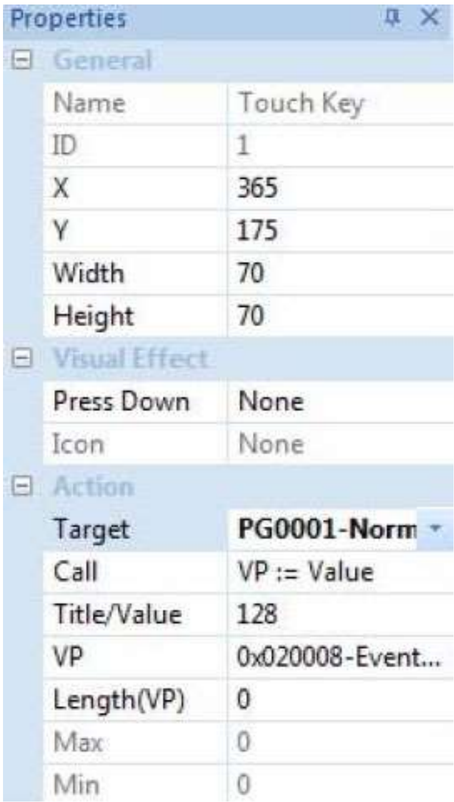
We now assign a Touch Key / Touch Area on to the On/Off key. This is done by placing it onto the On/Off button in the icon.



Note: M (Mode) and T (Tare) keys do not require touch areas in Off Type.

We set the properties as following:

- Location X = 365
- Location Y = 175
- Width = 70
- Height = 70
- Target: PG0001-Normal Type
- Call: VP := Value
- Title / Value = 128
- VP: 0x020008-Event Flags


Properties

General	
Name	Touch Key
ID	1
X	365
Y	175
Width	70
Height	70
Visual Effect	
Press Down	None
Icon	None
Action	
Target	PG0001-Norm
Call	VP := Value
Title/Value	128
VP	0x020008-Event...
Length(VP)	0
Max	0
Min	0

When setting the locations and size the Touch Key / Touch Area will adjust itself automatically. It is now placed on top of our On/Off area of our icon.

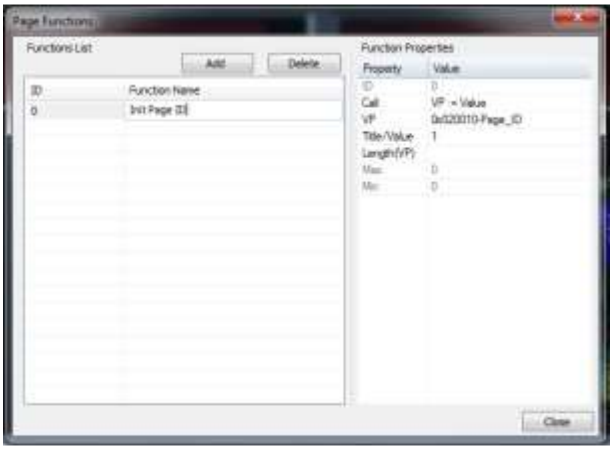
When a user presses the area of this Touch Key, then the module will transit to PG-0001-Normal Type and set bit 7 of Event Flags to 1, indicating to our Host MCU that the On/Off key is pressed. The Host MCU can read it via serial Interface and can reset it after processing the event.

We do not need to place any other touch areas or any other elements onto this page. Therefore we can proceed to designing PG0001-Normal Type

As for Page PG0001-Normal Type we should set initial values first.

However only one initial value is Required: Page_ID should be set to 1 any time this page is called.

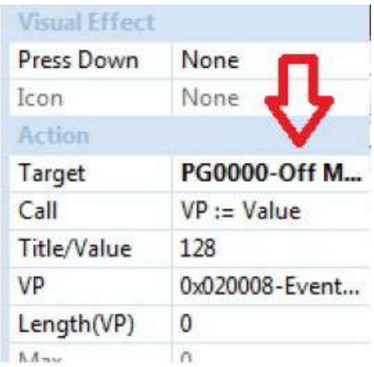
This indicates to Host MCU that we are currently in Normal Type.



ID	Function Name
0	Init Page ID

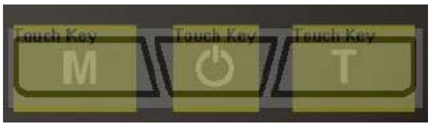
Property	Value
ID	0
Call	VP := Value
VP	0x020010-Page_ID
Title/Value	1
Length(VP)	
Max	0
Min	0

Now we are placing again our icon DI0000-keys2 to PG0001-Normal Type, and also our Touch Key for the On/Off Button. This time the target page for the Touch Key on the On/Off Button is PG0000-Off Type. All other values are the same as on PG0000-Off Type.



Visual Effect	
Press Down	None
Icon	None
Action	
Target	PG0000-Off M...
Call	VP := Value
Title/Value	128
VP	0x020008-Event...
Length(VP)	0
Max	0

On this page we also require two more touch areas for Mode and Tare:

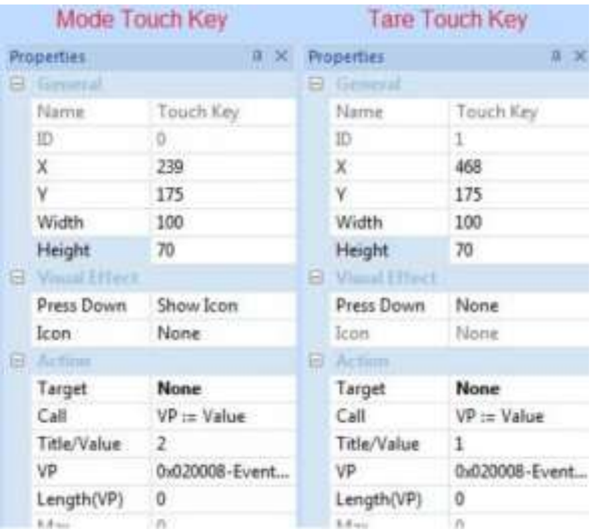


The properties of these two new Touch Keys / Touch Areas are as follow:

Only differences between the two areas are:

- X Location
- Event Flag should be set to 1 if Tare key is pressed， and to 2 if Mode key is pressed

Pressing the Tare key will set bit 0 of our VP Event Flags to 1 telling Host MCU that it should provide tare weight. After processing the flag, MCU should reset it via a Serial Communication VP write command.



Mode Touch Key

Properties

⊟ General	
Name	Touch Key
ID	0
X	239
Y	175
Width	100
Height	70
⊟ Visual Effect	
Press Down	Show Icon
Icon	None
⊟ Action	
Target	None
Call	VP := Value
Title/Value	2
VP	0x020008-Event...
Length(VP)	0
Max	0

Tare Touch Key

Properties

⊟ General	
Name	Touch Key
ID	1
X	468
Y	175
Width	100
Height	70
⊟ Visual Effect	
Press Down	None
Icon	None
⊟ Action	
Target	None
Call	VP := Value
Title/Value	1
VP	0x020008-Event...
Length(VP)	0
Max	0

Pressing the Mode key will set bit 1 of our VP Event Flags to 1, informing Host MCU that it should toggle the Units (gram or ounce).

Similarly, after processing the flag MCU need to reset it via a Serial Communication VP write command.

All three Touch Keys react independently on user interaction. Make sure you leave a small gap between those three touch areas.

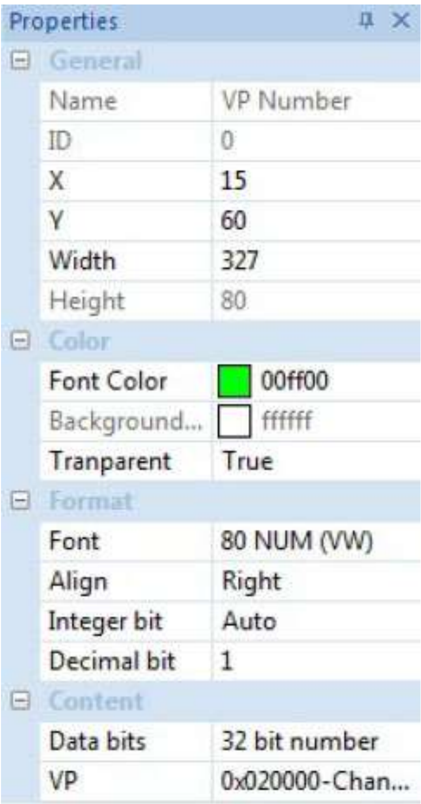
In Normal Type, the weight values from both load cell channels should be shown. We will place each of them into a VP Number Area onto our Page PG0001-Normal Type.

The left channel VP area will look as following and has the following properties:

- Location X = 15
- Location Y = 60
- Width = 327
- Height = 80 (defined by Font)
- Font Color = Green
- Transparent
- Font = 80 NUM (VW)
- Alignment: Right
- Auto Integer Digits
- 1 Decimal Digit
- VP = 32 Bit VP 0x020000-Channel1

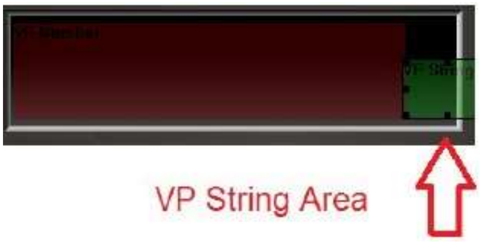
Now we insert the right channel VP area. Basically it will be the same except that X = 417 and the VP assigned to the area is VP 0x020004-Channel2

Beside the left channel value VP we should have a VP String area, which shall show “g” or “oz” depends on which unit is currently selected.



Properties

⊟ General	
Name	VP Number
ID	0
X	15
Y	60
Width	327
Height	80
⊟ Color	
Font Color	🟩 00ff00
Background...	□ ffffff
Tranparent	True
⊟ Format	
Font	80 NUM (VW)
Align	Right
Integer bit	Auto
Decimal bit	1
⊟ Content	
Data bits	32 bit number
VP	0x020000-Chan...

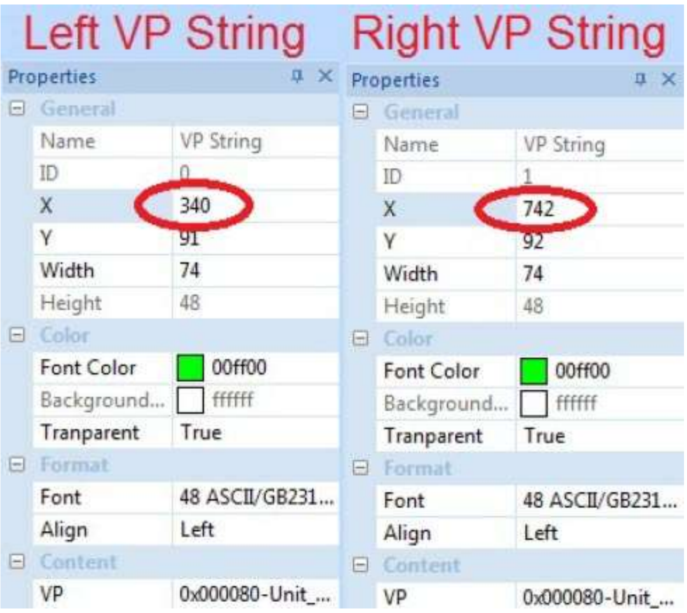


The properties of the two different String Areas only differ in their X locations:

We choose a Font with 48 pixels in height.

The VP assigned to the two VP String areas is 0x000080-Unit_String, which may contain the short strings “g” or “oz”.

The Host MCU has to set the correct string in VP 0x000080-Unit_String, when the unit is changing (e.g. the Mode Button is pressed).



Left VP String

Properties

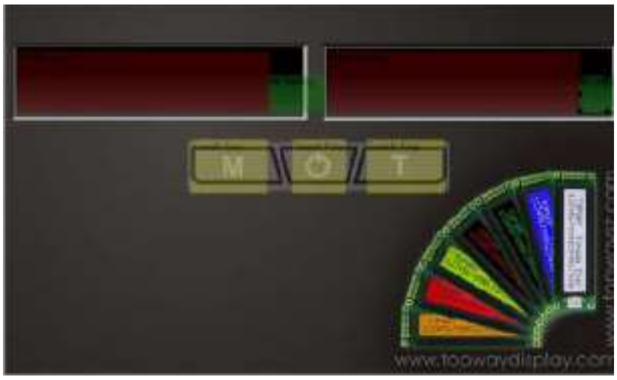
General	
Name	VP String
ID	0
X	340
Y	91
Width	74
Height	48
Color	
Font Color	🟩 00ff00
Background...	□ ffffff
Tranparent	True
Format	
Font	48 ASCII/GB231...
Align	Left
Content	
VP	0x000080-Unit_...

Right VP String

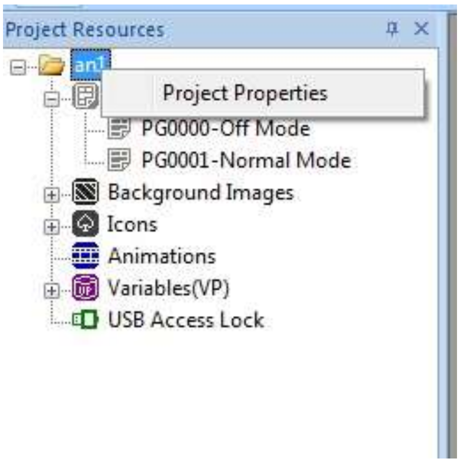
Properties

General	
Name	VP String
ID	1
X	742
Y	92
Width	74
Height	48
Color	
Font Color	🟩 00ff00
Background...	□ ffffff
Tranparent	True
Format	
Font	48 ASCII/GB231...
Align	Left
Content	
VP	0x000080-Unit_...

Now we have created the entire Page 2 and our project work is basically completed:



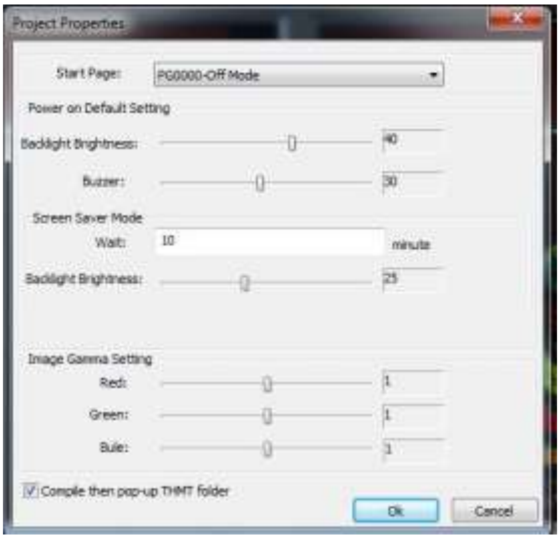
However before we upload the design to Smart LCD Module, we can set some initial project properties. Right click on the project will bring up its properties window.



Project Resources

- an1
  - Project Properties
  - PG0000-Off Mode
  - PG0001-Normal Mode
- Background Images
- Icons
- Animations
- Variables(VP)
- USB Access Lock

→



Project Properties

Start Page: PG0000-Off Mode

Power on Default Setting

Backlight Brightness: 40

Buzzer: 30

Screen Saver Mode

Wait: 10 minute

Backlight Brightness: 25

Image Gamma Setting

Red: 1

Green: 1

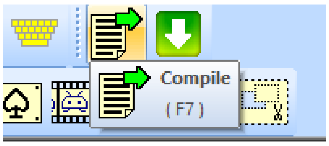
Bule: 1

☑ Compile then pop-up THMT folder

Here we can set the initial LCD Backlight intensity and the initial Buzzer timing. The most important thing is to select initial page. In this project, it is Page PG0000-Off Type.

Now we are ready to upload our project to HMT050ATA-2C.

1.) Disconnect any power from our HMT050ATA-2C (USB will supply power to the module)
2.) Connect the HMT050ATA-2C via the USB Interface to your PC
3.) The HMT050ATA-2C becomes visible to you as “Removable Disk”
4.) Pressing the “Compile” Button to generate project’s Output Files and open a Window to allow the user copy output files to HMT050ATA-2C (Removal Disk).



5.) Proceed dragging / copying the entire directories “Font” and “THMT” and accept overwriting any existing files.

6.) After copying these files to the module, you can disconnect it from computer.
7.) Connect the module with external power supply and power it on. Page PG00000-Off Type should become visible

2.7 How does Host MCU communicate with the HMT050ATA-2C and control it

HMT050ATA-2C connect to Host MCU via a Serial Interface cable. We will now discuss the general program flow inside the Host MCU and all necessary communication commands to read and write variables (VP).

Basically, the Host MCU is frequently reading the variable Page_ID to determine the system is in which operational type. If it is in Normal Type it should sample the ADC and calculate weight.

Furthermore, it will check the Event Flags and set Tare values if the Tare button is set or toggle the unit (gram or ounce) if the Mode Button is pressed.

Finally, the Host MCU transfers and stores the Weights to two VPs inside the HMT050ATA-2C, which will be displayed automatically by smart LCD module.

The Host MCU program flow may be as following:

After reset, the Host MCU will initialize internal clock, the SPI bus, the UART, a timer, the AD7190 and some internal variables.

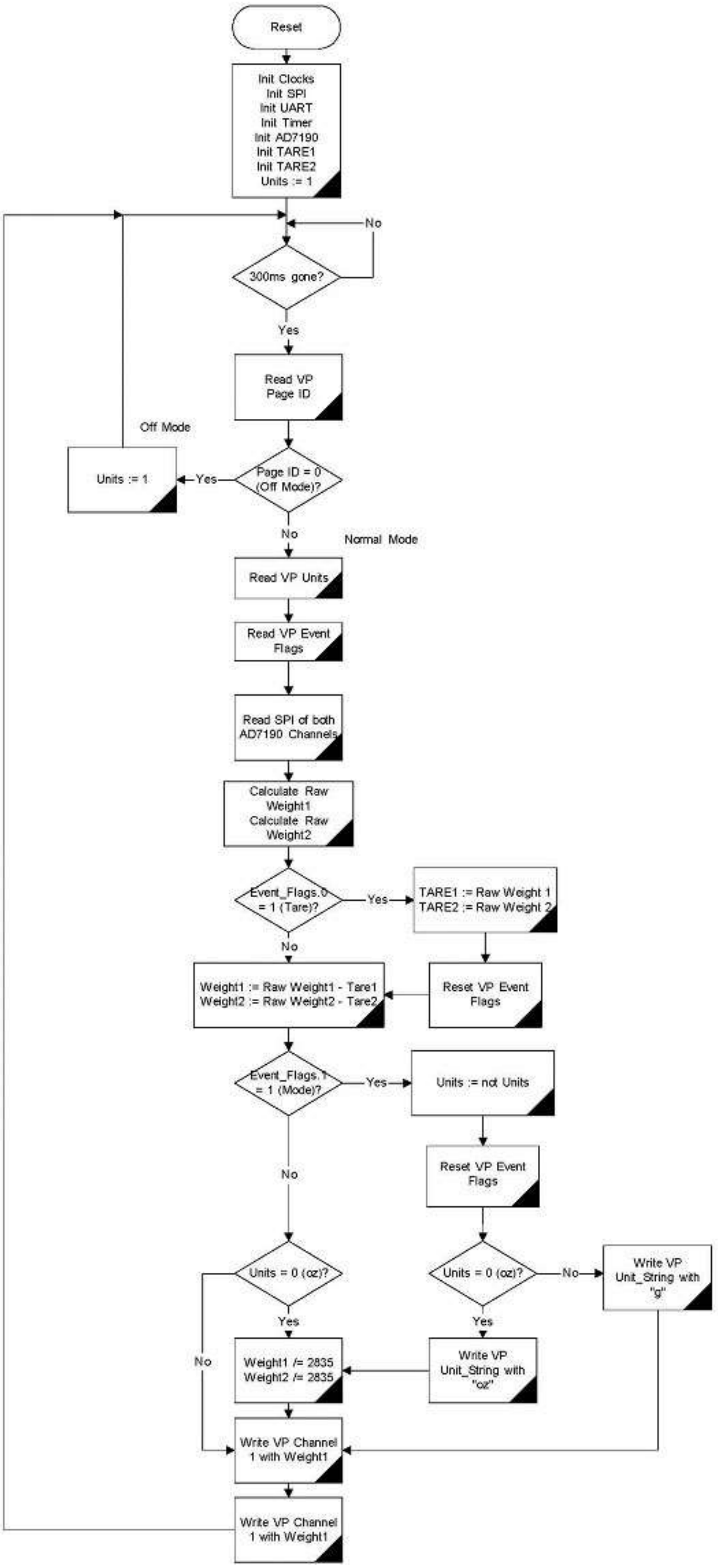
Every 300ms the Host MCU should start processing data.

First it reads the VP Page ID, which we placed at address 0x020010. If the current page is set to 0 then the scale operation mode is Off Type. The MCU just needs to reset Unit variable to 1(gram), since while running in Off Type we always set the units to gram.

If the VP Page ID is not 0 then we are in Normal Type. MCU should now read the VPs Units (address 0x020000C) and Event Flags (address 0x0200008). Furthermore, MCU queries the AD7190 to get the current voltage readings of two ADC channels. Based on the voltages we can calculate Raw Weights 1 and 2.

If the Tare Event Flag is set then We should record the new Tare 1 and Tare 2 values. The current Tare values are deducted from the two Raw Weights and save result in Weight1 and 2.

If the Mode Event Flag is set, we Need to toggle the Unit, store the corresponding string “g” or “oz” to string VP Unit_String (address 0x000080).



In case of current unit is ounce, we need to divide the Weights by 2835 to get values in ounce instead of gram. At last, we can store calculated values into two VPs representing Channel 1 (address 0x020000) and Channel 2 (address 0x020004).

Now we will describe the communication commands we use to communicate with Smart LCD Module HMT050ATA-2C.

1.) Read VP “Page ID”

The VP “Page ID” is placed at address 0x020010. Host MCU will transmit a so called VP Function to Smart LCD Module asking for the reading. In the datasheet of HMT050ATA-2C this command is called N32_read and uses the command code 0x45.

Before issuing command 0x45 (N32_read) we need to compose a data packet, which includes

- 1 Byte Packet Header (0xAA)
- 1 Byte Command Code (0x45)
- 4 Byte VP address (0x00, 0x02, 0x00, 0x10) with MSB first!
- 4 Byte Packet tail (0xCC, 0x33, 0xC3, 0x3C)

Therefore the entire 10-Byte package to request the value of Page ID would be:

- 0xAA, 0x45, 0x00, 0x02, 0x00, 0x10, 0xCC, 0x33, 0xC3, 0x3C

After receiving this package, HMT050ATA-2C will decode it and should return a package containing 32 Bit contents of VP Page ID. However before sending back data package the module would send a 2-Byte acknowledgment message 0x3A, 0x3E. Which we may or may not discard in Host MCU communication routine.

Important is the 10-Byte Response Package from our module:

- 1 Byte Packet Header (0xAA)
- 1 Byte Command Code on which this response is based (0x45)
- 4 Byte VP contents (MSB first!), in our case:
  - 0x00, 0x00, 0x00, 0x00 for PG0000-Off Type
  - 0x00, 0x00, 0x00, 0x01 for PG0001-Normal Type
- 4 Byte Packet tail (0xCC, 0x33, 0xC3, 0x3C)

The Host MCU should look for the 0xAA header with a following 0x45. If the bytes 7 … 10 are equal to the packet tail we can accept the packet as valid and accept the value for Page ID.

In our case the two possible (valid) response packages would be:

- 0xAA, 0x45, 0x00, 0x00, 0x00, 0x00, 0xCC, 0x33, 0xC3, 0x3C (Page ID is 0)
- 0xAA, 0x45, 0x00, 0x00, 0x00, 0x01, 0xCC, 0x33, 0xC3, 0x3C (Page ID is 1)

2.) Read VP “Units”

The VP “Units” is placed at address 0x02000C. We have to transmit again a N32_read command with code 0x45. The data package is as following

- 1 Byte Packet Header (0xAA)
- 1 Byte Command Code (0x45)
- 4 Byte VP address (0x00, 0x02, 0x00, 0x0C) with MSB first!
- 4 Byte Packet tail (0xCC, 0x33, 0xC3, 0x3C)

Therefore the entire 10-Byte package to request the value of Page ID would be:

- 0xAA, 0x45, 0x00, 0x02, 0x00, 0x0C, 0xCC, 0x33, 0xC3, 0x3C

After receiving this package, the HMT050ATA-2C will decode it and should return another package containing the 32 Bit contents of VP Units.

The 10-Byte Response Package from our module:

- 1 Byte Packet Header (0xAA)
- 1 Byte Command Code on which this response is based (0x45)
- 4 Byte VP contents (MSB first!), in our case:
  - 0x00, 0x00, 0x00, 0x00 for Ounce
  - 0x00, 0x00, 0x00, 0x01 for Gram
- 4 Byte Packet tail (0xCC, 0x33, 0xC3, 0x3C)

The Host MCU should look for the 0xAA header with a following 0x45. If the bytes 7 … 10 are equal to the packet tail we can accept the packet as valid and accept the value for Units.

In our case the two possible (valid) response packages would be:

- 0xAA, 0x45, 0x00, 0x00, 0x00, 0x00, 0xCC, 0x33, 0xC3, 0x3C (Units is Ounce)
- 0xAA, 0x45, 0x00, 0x00, 0x00, 0x01, 0xCC, 0x33, 0xC3, 0x3C (Unit is Gram)

3.) Read VP “Event Flags”

The VP “Event Flags” is placed at address 0x020008. Host MCU again transmits a N32_read command with code 0x45. The data package is as following

- 1 Byte Packet Header (0xAA)
- 1 Byte Command Code (0x45)
- 4 Byte VP address (0x00, 0x02, 0x00, 0x08) with MSB first!
- 4 Byte Packet tail (0xCC, 0x33, 0xC3, 0x3C)

  Therefore the entire 10-Byte package to request the value of Event Flags would be:
- 0xAA, 0x45, 0x00, 0x02, 0x00, 0x08, 0xCC, 0x33, 0xC3, 0x3C

After receiving this package, the HMT050ATA-2C will decode it and should return another package containing the 32 Bit contents of VP Event Flags.

The 10-Byte Response Package from our module:

- 1 Byte Packet Header (0xAA)
- 1 Byte Command Code on which this response is based (0x45)
- 4 Byte VP contents (MSB first!), in our case:
  - 0x00, 0x00, 0x00, 0x00 for no event Flag set
  - 0x00, 0x00, 0x00, 0x01 for Tare Flag set
  - 0x00, 0x00, 0x00, 0x02 for Mode Flag set
  - 0x00, 0x00, 0x00, 0x80 for On/Off Flag set
- 4 Byte Packet tail (0xCC, 0x33, 0xC3, 0x3C)

The Host MCU should look for the 0xAA header with a following 0x45. If the bytes 7 … 10 are equal to the packet tail we can accept the packet as valid and accept the value for Event Flags.

In our case the two possible (valid) response packages would be:

- 0xAA, 0x45, 0x00, 0x00, 0x00, 0x00, 0xCC, 0x33, 0xC3, 0x3C (No Flag set)
- 0xAA, 0x45, 0x00, 0x00, 0x00, 0x01, 0xCC, 0x33, 0xC3, 0x3C (Tare Flag set)
- 0xAA, 0x45, 0x00, 0x00, 0x00, 0x02, 0xCC, 0x33, 0xC3, 0x3C (Mode Flag set)
- 0xAA, 0x45, 0x00, 0x00, 0x00, 0x80, 0xCC, 0x33, 0xC3, 0x3C (On/Off Flag set)

4.) Reset VP “Event Flags”

The VP “Event Flags” is placed at address 0x020008. Now host MCU sends a N32_write command with code 0x44. We will write 0x00000000 to the VP address to reset the contents of the VP. The data package is as following

- 1 Byte Packet Header (0xAA)
- 1 Byte Command Code (0x44)
- 4 Byte VP address (0x00, 0x02, 0x00, 0x08) with MSB first!
- 4 Byte 32 Bit Data (0x00, 0x00, 0x00, 0x00) with MSB first!
- 4 Byte Packet tail (0xCC, 0x33, 0xC3, 0x3C)

Therefore the entire 14-Byte package to reset the value of Event Flags would be:

- 0xAA, 0x44, 0x00, 0x02, 0x00, 0x08, 0x00, 0x00, 0x00, 0x00, 0xCC, 0x33, 0xC3, 0x3C

After receiving this package, the HMT050ATA-2C will decode it, set the VP to 0 and should return just an acknowledgement message with 0x3A, 0x3E

5.) Write VP "Unit String"

The VP "Unit String" is placed at address 0x000080. Host MCU will transmit a STR_write command with code 0x42. The data package is as following

- 1 Byte Packet Header (0xAA)
- 1 Byte Command Code (0x42)
- 4 Byte VP address (0x00, 0x00, 0x00, 0x80) with MSB first!
- 2 or 3 Byte 0-Terminated String Data for "oz" or "g"
- 4 Byte Packet tail (0xCC, 0x33, 0xC3, 0x3C)

Therefore the entire package to set Unit String would be:

- 0xAA, 0x42, 0x00, 0x00, 0x00, 0x80, 0x6F, 0x7A, 0x00, 0xCC, 0x33, 0xC3, 0x3C (oz)
- 0xAA, 0x42, 0x00, 0x00, 0x00, 0x80, 0x67, 0x00, 0xCC, 0x33, 0xC3, 0x3C (g)

After receiving this package, the HMT050ATA-2C will decode it, set the String VP to "oz" or "g" and should return a short acknowledgement message 0x3A, 0x3E

6.) Write VP "Channel1"

The VP "Channel1" is placed at address 0x0200000. Now we transmit a N32_write command with code 0x44 and save a 32 Bit number representing the weight of Channel 1. The data package looks as following

- 1 Byte Packet Header (0xAA)
- 1 Byte Command Code (0x42)
- 4 Byte VP address (0x00, 0x02, 0x00, 0x00) with MSB first!
- 4 Byte signed 32 Bit Data with MSB first. These data represent the weight of Channel 1 in 0.1g or 0.1oz steps
- 4 Byte Packet tail (0xCC, 0x33, 0xC3, 0x3C)

Therefore the entire 14-Byte package to reset the value of Channel1 would be:

- 0xAA, 0x44, 0x00, 0x02, 0x00, 0x00, 0xnn, 0xnn, 0xnn, 0xnn, 0xCC, 0x33, 0xC3, 0x3C

After receiving this package, the HMT050ATA-2C will decode it, set the VP with the desired weight value and should return a acknowledgement message with 0x3A, 0x3E

7.) Write VP "Channel2"

The VP "Channel2" is placed at address 0x0200004. Similarly, the 14-Byte package to reset the value of Channel2 would be:

- 0xAA, 0x44, 0x00, 0x02, 0x00, 0x04, 0xnn, 0xnn, 0xnn, 0xnn, 0xCC, 0x33, 0xC3, 0x3C

After receiving this package, the HMT050ATA-2C will decode it, set the VP with the desired weight value and should return a acknowledgement message with 0x3A, 0x3E

The Smart LCD Module HMT050ATA-2C provides a far more advanced command set than we have used in this project. We only focused on commands needed for this specific application.

3. Summary

To recap, we covered the followings:

- The general principle of using Smart LCD Module
- How to use Topway's UI Editor
- The basic Serial Communication between a Host MCU and the Smart LCD Module
- A few additional information about host MCU and Data Acquisition

The Application Note is to jumpstart product designer on using Topway's Smart LCD Module. It shows the unique features and capabilities of Smart LCD Module and its advantages like:

- Very quick UI design turnover because of advance RG Tools Editor, without the need of low level programming on pixel drawing
- Releasing host MCU from most of the UI tasks
- Extending the limited resource of host MCU such as
  - Additional Data Flash
  - Additional Data RAM
  - A Real Time Clock