HMC-20

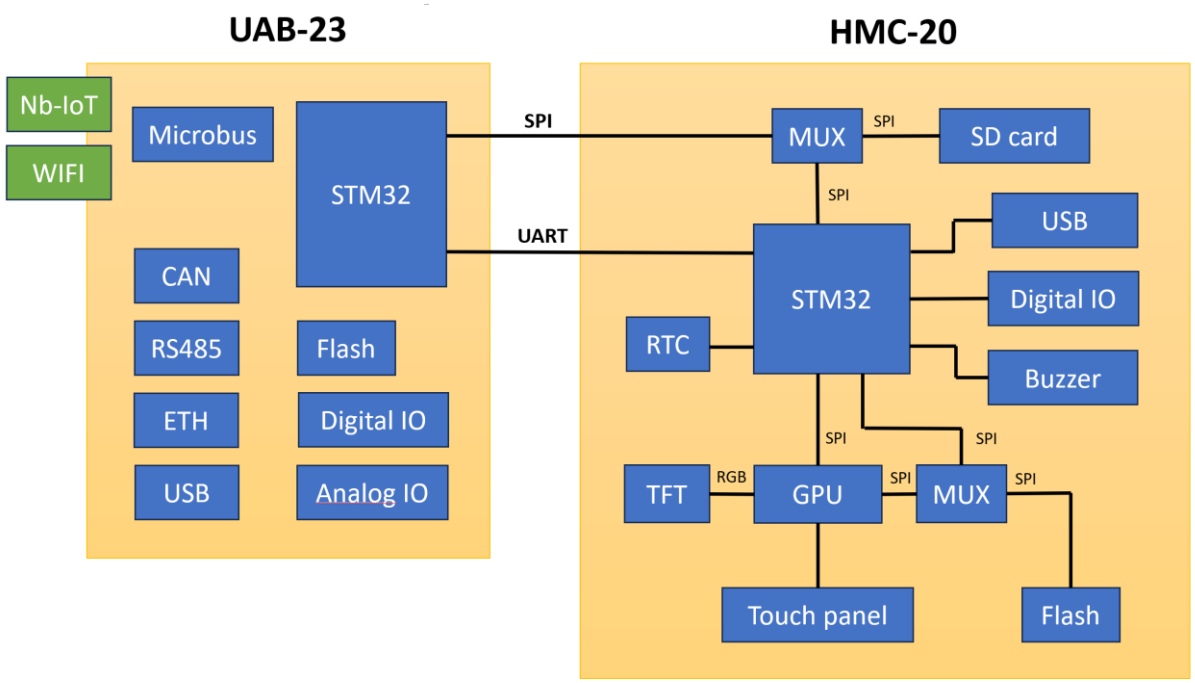
7” Highly configurable Human Machine Controller

Revision: 18/04/2024

Content

1. Introduction

The HMC-20 *Human Machine Controller* in combination with the UAB-23 *User Application Board* forms a multi-functional and fully configurable Human Machine Interface console that can be used for a wide range of applications. The split-up between a graphical and an application part makes it possible to use the console in different environments with a dedicated board with specific IO for user defined purposes. Both boards are available separately, and the UAB-23 design is open source for further customization. A case is available to mount both boards in a stackable way with invisible cabling. The case is available in multiple colors and different sizes. Available display sizes are 3,4" and 7". There is auditory feedback that produces different sound effects when touching buttons. The HMC-20 contains 5 digital inputs, 5 digital outputs with selectable voltage, an SD-card interface, USB2.0 interface, RTC with battery, power interface and a user application board connector. The TFT display has a capacitive or resistive touch interface and can be dimmed with PWM or put in sleep mode. The layout, interactions, images and widgets are configurable by the user by putting them on the SD-flash card as BMP/JPG images. Properties and UI-behavior is defined in a Json file. The application board connector contains a real time command interpreter UART interface to drive the graphical part with easy, human readable, textual commands.

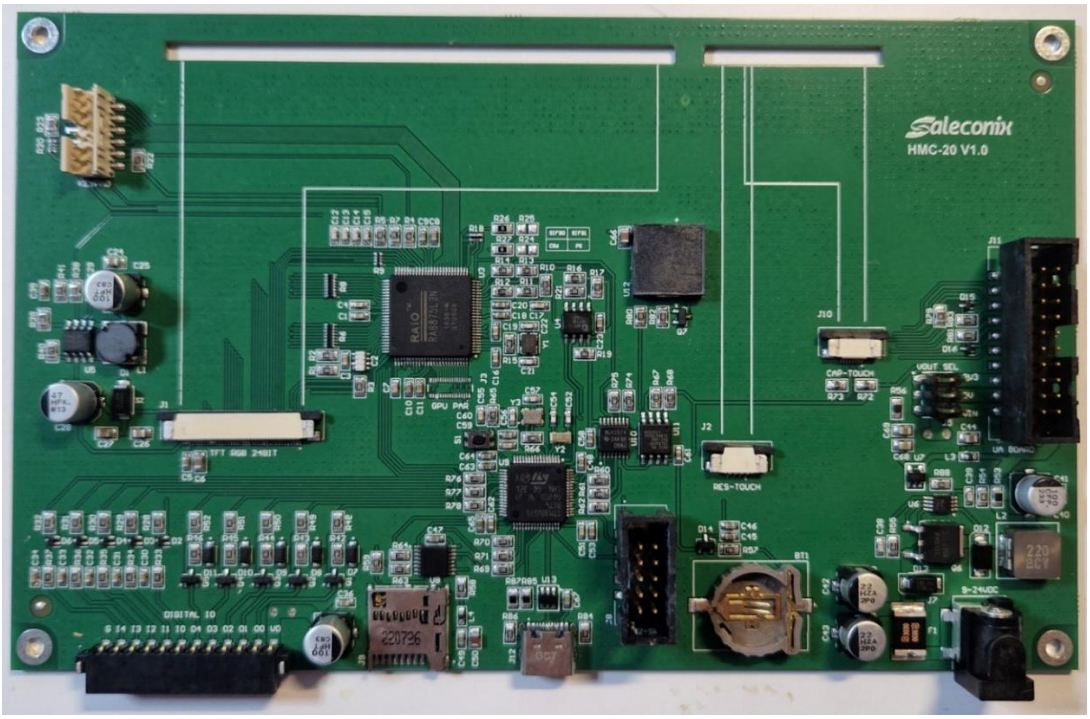


2. Application Domains

- General purpose HMI with static or animated graphics
- Agriculture (greenhouse, grow room)
- Buildings (domotics, liquid levels, syndic)
- Automotive, tank levels, etc.
- Industrial applications

3. Hardware Features

HMC-20 board contains the graphical part that consists of the following parts (from left to right):



1. 7” 24bit 800x480pixels RGB TN TFT display, 1000cd/m2 with 60 degree viewing angle.

2. Keypad interface (J4) to attach a keyboard with up to 4x5 physical buttons in a matrix pattern to be used as an alternative for the touch panel interface.

3. PWM controlled LED backlight convertor generates 10V/300mA boost conversion for high brightness TFT display.

4. 5 digital Inputs and 5 digital Outputs with voltage selector. Internal pullup allows reading buttons or driving signals. Protected by diodes to drive relays or voltage driven LED-strips.

5. 24bit RGB FPC connector for TFT display, includes connections for backlight, video synchronization and resistive touch interface.

6. Graphics Processing Unit (GPU) allows putting graphical shapes, fonts and images on the display. 256/65K colors, up to 800x480 pixels. Internal and user defined font ROM 8x16 dots with enlargement and rotation function.

7. Central Processing Unit (CPU) with external clock, reset button, programming connector and high accuracy Real Time Clock (RTC). Acts as the central controller between all interfaces and GPU.

8. SD-card interface to import BMP/JPG images into board flash and read the layout. Once copied, the flash card can be removed.

9. EMC protected USB2.0 interface to access flash card as a drive or to perform a firmware update. Contains also a debug interface.

10. Font ROM up to 15x16 dots[1], GB12345/BIG5 (Chinese) and JIS0208 (Japanese) character set. Language support for 150 countries including Latin, Cyril, Greek and Arabian.

11. 64Mb serial NOR flash to save images, layout and data with optimized index and HMAC (Hash-based message authentication code) for security purposes.

12. Piezo electric buzzer for auditory feedback allows the generation of different sound effects when pressing buttons.

13. Resistive and capacitive touch interface for touch control from the TFT panel. The capacitive touch data is read directly from the CPU. Support for gesture and Hotknot (capacitive transmission of data between screens).

14. Holder for 3V battery to keep real time clock running when no external power source is available.

15. User Application board interface connector (J11) to send serial commands to command interpreter in human readable format. Compatible with 3V3 FTDI serial to USB convertor cable. Contains access lines to flash card to store files from an external source.

16. Power supply that generates 5V at max 4A from a wide voltage input range 9-24VDC.

[1] Other font chips are possible up to 32x32 dots. Contact us for further support.

4. Software Features

- 9 predefined visible/non-visible widgets that are fully configurable and assignable to pages
- Real time command interpreter to send and receive sequences defined as widget events or from UART to modify properties and render pages
- Direct Memory Access (DMA) interface to store images from SD-card and save data on flash chip with indexing mechanism and HMAC to ensure data integrity
- Layout importing mechanism from Json file[2]
- Calibration mechanism for resistive touch displays
- Initialization procedures
- Real time clock setting mechanism
- Rotation function: 0, 90, 180, 270°
- Remote firmware updates[3]
- Extended logging trough UART
- 3 demo apps: button, fogger and runlight with user application as a Python script

5. Specifications

5.1. Absolute Maximum Ratings

Parameter	Min	Typ	Max	Unit
Input Supply Voltage	-0.3	-	36	V
Operating Temperature	-20	-	70	°C
Storage Temperature	-30	-	85	°C
Storage Humidity	20	-	90	%RH
Digital Input Voltage	-36,7	-	40	V

5.2. Electrical Characteristics

Parameter	Min	Typ	Max	Unit
Input Supply Voltage	4,5	-	35	V
Maximum Power consumption[4]	TBD	-	4	W
Digital Input Vth Low	-	-	1,254	V
Digital Input Vth High	1,85	-	-	
Digital Output Voltage[5]	-	3,3/5/Vin	-	
Maximum output current	-	-	1,6	A
Maximum total output power @5V	16[6]	-	20	W
Maximum total output power @3V3	-	-	0,825[7]	
Serial port speed	-	67800	-	Bd

[2] Feature not implemented yet. Contact us for layout support.
[3] From user application board as a file to SD card.
[4] When digital output is not delivering power
[5] Selectable with J5
[6] When PWM display is at 100%
[7] When PWM display is at 100%

5.3. Optical Characteristics

Parameter	Min	Typ	Max	Unit
Brightness	900	1000	-	cd/m²
Horizontal Viewing Angle	60	70	-	degree
Vertical Viewing Angle Top	40	50	-	
Vertical Viewing Angle Bottom	60	70	-	
Contrast ratio	400	500	-	-
Uniformity	-	80 (If=270mA)	-	%
Life Time	-	50000 (If=270mA)	-	Hour
Response Time Rising	-	10	20	ms
Response Time Falling	-	15	30	
CF Color CIE 1931	-0,05	0,05	-	-
Backlight Color	White			
Colors	256			
Bit per pixel	8			

5.4. Physical Characteristics

Parameter	Min	Typ	Max	Unit
TFT size	-	7	-	Inch
Active display area	-	154,08(W)x85,92(H)	-	mm
Dot pitch	-	0,1926(W)×0,1790(H)	-	
Display dimension	-	164,9(W)x100(H)	-	
Board dimension inc. display	-	179,9(L)x111(W)x22(H)	-	
Mounting Hole Size	-	3	-	
Weight	-	TBD	-	g

6. Connections

6.1. Power interface

Pin	Function
Central pin	Positive power supply input
Border contact	Negative power supply input

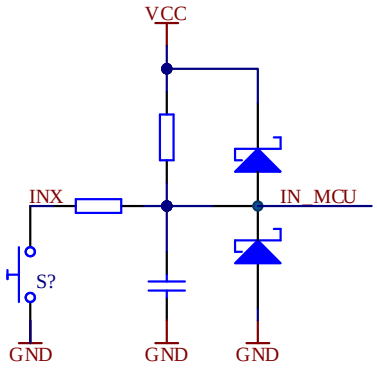
6.2. IO interface – J6

Pin number	1	2	3	4	5	6	7	8	9	10	11	12
Function	G	I4	I3	I2	I1	I0	O4	O3	O2	O1	O0	VO

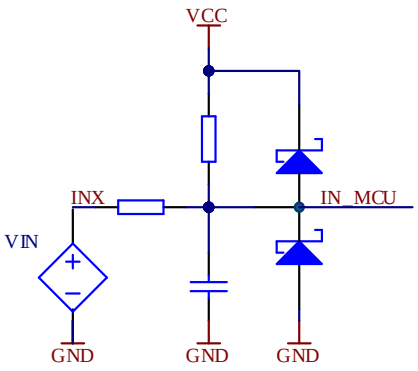
Function	**Info**	**Function**	**Info**
G	Ground	O4	Output4
I4	Input4	O3	Output3
I3	Input3	O2	Output2
I2	Input2	O1	Output1
I1	Input1	O0	Output0
I0	Input0	VO	Output voltage

Input configurations:

The inputs are flexible, debounced and protected against spikes and overvoltage. There are two different configurations that can be used:



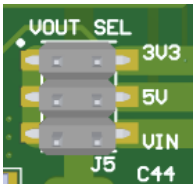
Input button configuration
with internal pullup resistor (100k)



Active input voltage

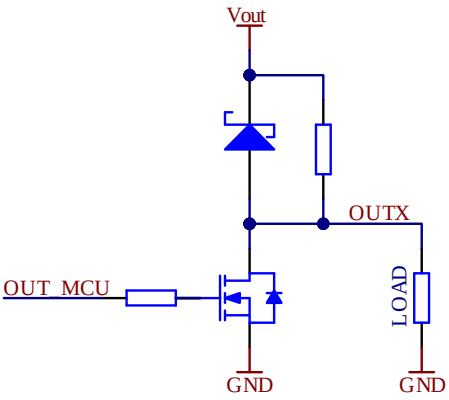
Output configurations:

The output voltage (VO) can be selected with jumper J5 as in following table:

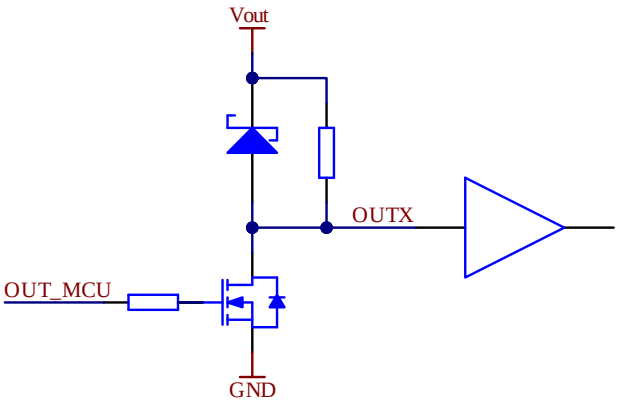


1-2	3V3 (=Vcc)
3-4	5V
5-6	VIN

The outputs are protected with a diode against inductive loads and can be used in the following configurations:



Driving a load (1,6A max)

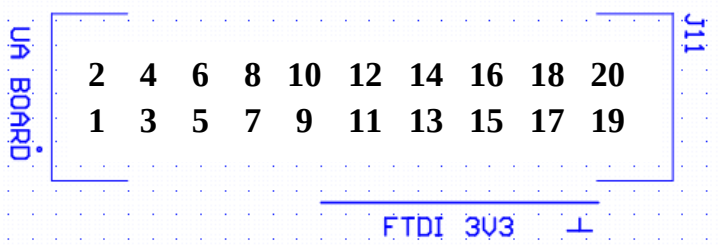


Digital output with internal pullup resistor (10k)

Using higher Vout values allows reduction of the current to drive led strips, etc. The total power delivered by all outputs may not exceed 16W.

6.3. UA board interface – J11

The user application board interface contains a serial port to send and receive commands to and from the command interpreter and SPI access to the SD flash card to store files from an external source.



Pin number	Function	Info	Pin number	Function	Info
1	Vin	Input Power voltage	2	Vin	Input Power voltage
3	5V	Regulated voltage	4	5V	Regulated voltage
5	3V3		6	3V3	
7	NC		8	NC	
9	NC	Command interface 3V3 FTDI cable compliant	10	NC	
11	Rx		12	MOSI	SD card interface
13	Tx		14	CLK	
15	NC		16	MISO	
17	NC		18	SDSEL	
19	GND		20	GND	

7. Real time interpreter commands

There are three formats for commands that the interpreter recognizes:

Property assignment for widget properties:

```
[pagename].[widget].[propertyname]=[property]\n
```

Property assignment for page properties:

```
[pagename].[propertyname]=[property]\n
```

Command for general actions:

```
[command] [argument]\n
```

Every command is ended with a newline character **'\n'**. This makes it also possible to enter the commands in command line for testing purposes. Some examples are:

```
"code.unlockkeypad.cmd='exe'\n" // Unlock the keypad
“code.enter.act=true\n"         // Unlock the enter button
"code.crosshome.act=true\n"     // Unlock the cross-home button
"code.ok.vis=false\n"           // Make the text 'ok' invisible
"code.pincodesaved.vis=false\n" // Make the text 'pincodesaved' invisible
"code.popup.vis=false\n"        // Make the green popup window invisible
"page 'code'"                   // Redraw the code page
```

Properties and arguments can be of the type integer, Boolean or string. Strings are always quoted with single quotes. Embedded strings are quoted again in the deeper level of the command and are escaped by a double escape character '\\' belonging to each level. For example, the following command:

```
"code.pincode.pe='code.enter.tpe=\\'code.pincode.cmd=\\\\'save\\\\'\\''\n"
```

Has the following meaning (read from left to right):

“Set the positive evaluation event (pe) of the pincode widget (pincode) on the code page (code) to update the touch press event (tpe) of the enter button (enter) on the code page (code) to save the pincode by executing a command (cmd) with argument 'save' on the pincode widget (pincode) on the code page (code).

A breakdown gives:

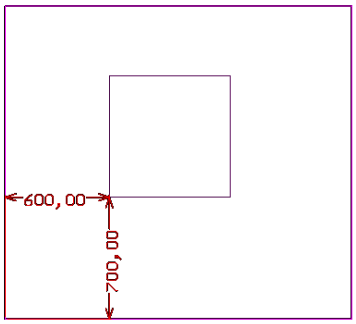
```
"code.pincode.pe='A'\n"             -> level 0 (no escapes)
A = code.enter.tpe= \\'B\\'         -> level 1 (double escapes)
B = code.pincode.cmd=\\\\'C\\\\'    -> level 2 (quadruple escapes)
C = save
```

The widgets are divided into visual and non-visual types. Every widget has to be assigned to a page and contains common or specific properties. Properties can be updated dynamically and made directly visible or in silent mode. In that case the properties are updated in the background and are only visible when a page render is executed of when the silent update mode is turned off.

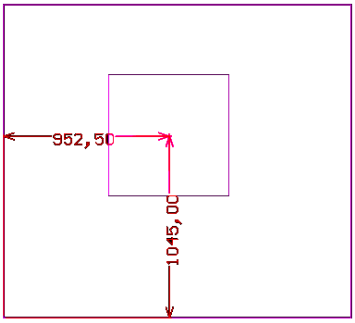
All widgets have a name (string) which is unique for each page as a reference handle and a type (integer) which identifies the widget type.

Property	**Info**	**Data type**	**Access type**
Name	The name which acts as a unique identifier	String	R/W
Type	The type number to identify the widget type	Integer	R

The position of the visual widgets is defined by coordinates x and y and a selected coordinate system. The corner coordinate system defines the coordinates of the widget in the left bottom corner of the widget canvas referenced to the left bottom of the screen. The center coordinate system defines the coordinates of the widget in the center point of the widget canvas references to the left bottom of the screen. Select the system that is most appropriate for the calculations.



Corner coordinate system



Center coordinate system

7.1. Visual widgets

Common widget properties:

Type	Property	Comment	Data type	Range	Command Identifier
Representation	Level	The higher the level, the higher the widget is layered on the screen	Integer		
	Background color	When the type is 'solid background', the color can be selected as an integer value	Integer		bco
	Image	When the type is 'image', the image can be selected by an identification number	Integer		img
	Image increment	When the image is used as a counter or an animation, the increment command renders the next image in the sequence			imi
	Image decrement	When the image is used as a counter or an animation, the decrement command renders the previous image in the sequence			imd
	Location x	The coordinates of the widget using the defined coordinate system	Integer		
	Location y				
	Size width	The size of the widget must be the same size of the background image	Integer		
	Size height				
	Extended width	The extended size defines the area	Integer		
	Extended height				
	Status	The status defines the status of the background. A solid button could be a confirmation button and a image button could be a touchable symbol.		'noback' 'solid' 'image'	
Behavior	Active	When the widget is not active, it will not react on touch events	Boolean Integer	true/false 0/1	act
	Border type	Show the border canvas of the widget	Boolean Integer	true/false 0/1	
	Border color	Define the color of the widget canvas as an integer value			
	Sound type	Define the sound type of the auditory interface when the widget is touch pressed			
	Sound enable	Enable or disable the touch press sound	Boolean Integer	true/false 0/1	
	Visibility	Make the widget visible or invisible on the screen	Boolean Integer	true/false 0/1	vis
	Coordinate system	Define the coordinate system between corner or center coordinates			
	Clockpart	Define the part of the clock when the widget is used to represent a time or a date	String	hourh hourl minh minl	
	Offset	Set the image offset for the clock part	Integer		
Event actions	Touch press	Define the touch press event	String	-	tpe
	Touch release	Define the touch release event	String	-	
	Touch up	Define the touch up gesture event	String	-	
	Touch down	Define the touch down gesture event	String	-	
	Touch left	Define the touch left gesture event	String	-	
	Touch right	Define the touch right gesture event	String	-	

Specific widget properties:

Widget	Type	Property	Comment	Data type	Range	Command Identifier
Page	Behavior	Silent update	When the silent update mode is on, the properties of the widgets coupled to that page are updated in the background to avoid transition effects			sil
Button						
Led						
Progress bar		Value	The value of the progress bar			val
Textual		Text	The text string of the textual			txt

7.2. Non visual widgets

General actions:

Action	Comment	Data type	Command Identifier
Page render	Render the page defined by a specific name	string	page
Send command	Send a command to the serial port		get
Print screen	Save the current screen on the flash card		pscr

Specific widget properties:

Widget type	Type	Property	Argument	Info	Data type	Range	Command Identifier
Timer	Behavior	Enable					en
		Start					
		Stop					
	Event action	Run out					
Pincode	Behavior	Command			String	‘eval’	cmd
	Event action	Pass event					pe
		Fail event					fe
		Saved successful					
Function	Behavior	Execute			String	‘exe’	cmd
Clock	Behavior	Set time/date					set

8. Json layout format

To implement

9. Layout update procedure

To implement

10. Firmware update procedure

To implement

11. Certifications

Certification procedure in progress

12. Order codes

Order code	Comment	Moq
HMC-20-C	Human machine controller PCB with capacitive touch panel	5
HMC-20-R	Human machine controller PCB with resistive touch panel	5
UAB-23	User application board with schematic and demo source code	5
HMC-CASE-3D38	Case to mount both PCB's 3D printed, height 38mm	1
HMC-CASE-38	Case to mount both PCB's with mold, height 38mm	500

Send a mail to info@saleconix.be for quotes and delivery times.