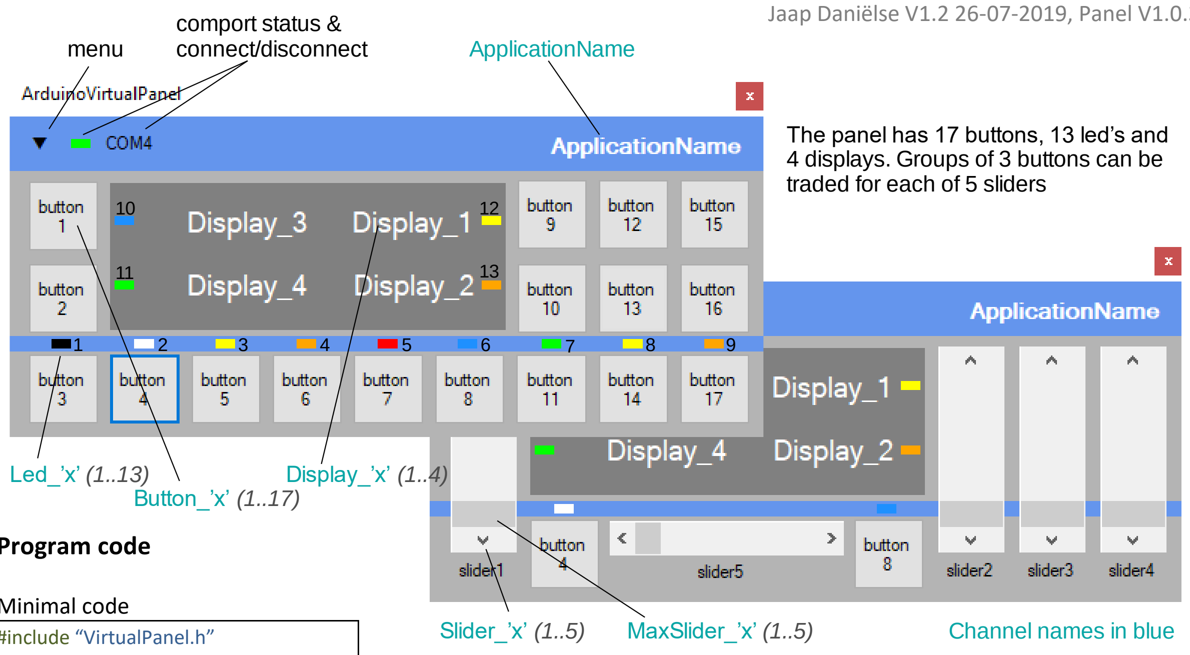
Virtual Panel

Quick Reference

Arduino Experiment Control Panel

Jaap Daniëlse V1.2 26-07-2019, Panel V1.0.3



The panel has 17 buttons, 13 led's and 4 displays. Groups of 3 buttons can be traded for each of 5 sliders

Program code

Minimal code

```
#include "VirtualPanel.h"

void setup ()
{   Panel.begin();          }

void loop ()
{   Panel.receive();   }

// callbackroutine
void PanelCalback (vp_channel event)
{ switch (event)
  {  case channel:
       …
     break;
} }
```

```
Panel.send (channel, variable*);
Panel.sendf (channel, formatted
string**, format variables ...);
```

* types depending on channel.
** see printf formatting

Main Panel channels/events

send() or sendf() functions.
Receive via PanelCallback() event

ApplicationName — send

string	application name text
color[1]	text color

PanelConnected — receive

void	On connect.

Reset — send

void	resets panel

DynamicDisplay — send

bool	activate/ deactivate
int16	delay mS (100-2000)*

Receive

void	on delay freq.

*Default 250mS

UnixTime — send

void	request

receive

uint32	(local) unix time code

Beep — send

void	500 Hz 400 ms
Int16	Freq. Hz 400 ms
uint32*	Frequency, Duration

* use _Sound helper function:
long _Sound(int freq, int dur)

Button_'x' (1..17) — send

any	button text
color[1]	button color
size[1]	text size

receive

void	on button click

Slider_'x' (1..5) — send

bool	visible/invisible
string	slider label text
int16	set (initial) value

receive

int16	value on slider action

MaxSlider_'x' (1..5) — send

int16	maximum value*

* positive only

[1] See: *Special strings*

Led_'x' (1..13) — send

bool	visible/invisible
color[1]	led color

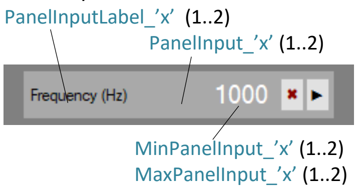
Display_'x' (1..4) — send

any	display as text
color[1]	display color
size[1]	text size

Receive

void	on double click

Panel Input



PanelInput_'x' (1..2) — send

bool	static/volatile
any	value

receive

any *	value

*Type same as sent type

MinPanelInput_'x' (1..2) — send

num*	value

MaxPanelInput_'x' (1..2) — send

num*	value

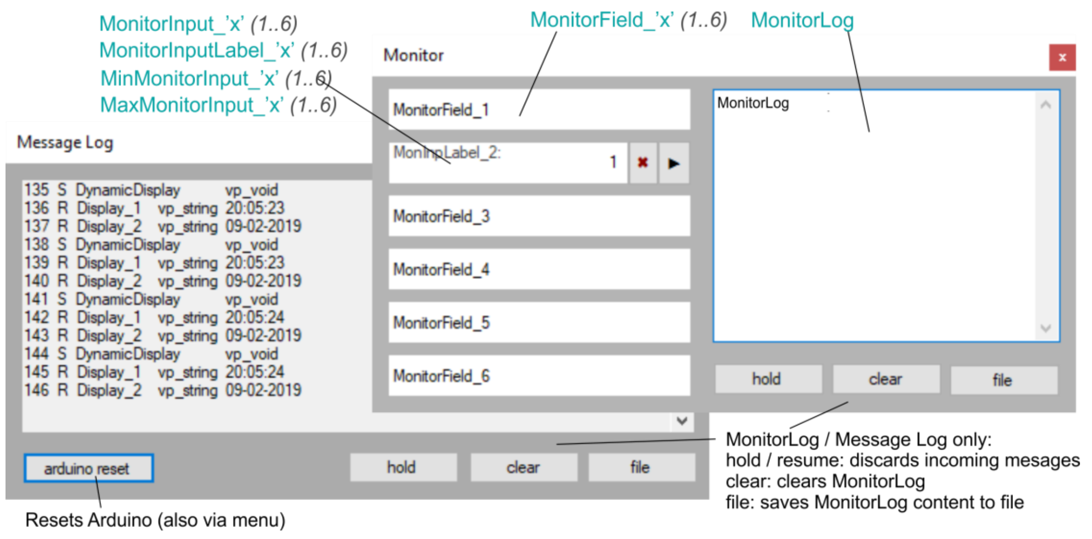
*Int16 int32, float32
When string min/max length.

PanelInputLabel_'x' (1..2) — send

any	Input label text

Message Log Panel
Records incoming (R) and outgoing (S) messages.

Monitor panel
Provides a log panel and additional displays and inputs



```
135  S  DynamicDisplay        vp_void
136  R  Display_1   vp_string  20:05:23
137  R  Display_2   vp_string  09-02-2019
138  S  DynamicDisplay        vp_void
139  R  Display_1   vp_string  20:05:23
140  R  Display_2   vp_string  09-02-2019
141  S  DynamicDisplay        vp_void
142  R  Display_1   vp_string  20:05:24
143  R  Display_2   vp_string  09-02-2019
144  S  DynamicDisplay        vp_void
145  R  Display_1   vp_string  20:05:24
146  R  Display_2   vp_string  09-02-2019
```

MonitorLog / Message Log only:
hold / resume: discards incoming mesages
clear: clears MonitorLog
file: saves MonitorLog content to file

Message Log

Format:
146 R Display_2 vp_string Test
{MessageNumber}{Send/Receive}
{channel}{VarType}{Value}

Monitor channels / events

Monitor   send

bool	win. visible/invisible

MonitorField_'x' (1..6)   send

any	display as text

MonitorInput_'x' (1..6)   send

bool	static/volatile
any*	value

receive

any*	value

*Type same as sent type

MonitorInputLabel_'x' (1..2)   send

any	Input label text

MinMonitorInput_'x' (1..6)   send
MaxMonitorInput_'x' (1..6)   send

num*	value

*Int16 int32, float32
When string min/max length.

MonitorLog   send

any	display as text
$CLEAR	clear Log

Special strings

Color strings
For: ApplicationName, Display, Led, Button.

$DELETE*	
$OFF**	⬛
$BLACK	⬛
$GRAY	▨
$PURPLE	🟪
$PINK	🟪
$BLUE	🟦
$GREEN	🟩
$YELLOW	🟨
$ORANGE	🟧
$RED	🟥
$BROWN	🟫
$WHITE	⬜

* draw only ** Led only

Graph Type strings
Set graph type. Rolling values are added right and move to left. Static waits until all values have been sent then displays.

$ROLING*	Set rolling graph
$STATIC	Set static graph

* default

Pen size strings Draw
GraphPen, GraphValue

$1PX*	1 pixel
$2PX	2 pixels
$3PX	3 pixels
$4PX	4 pixels

* default

Text attributes/size strings

$SMALL	fontsize small
$NORMAL*	fontsize normal
$BIG	fontsize big
$BOLD	bold text

*Default. Resets bold and big

Unicode characters
Using send() or sendf() to send a string, Unicode characters can be used. Simply copy and paste into the string.

Helper function Sound
long _Sound(int freq, int dur)
Combines two int16_t (frequency Hz, duration mS) into one uint32_t.

Helper functions Draw
uint16_t _Point(byte x, byte y)
combines 2 bytes into uint16_t (x,y) for a point.
When sent to GraphDrawLine consecutive points are connected in a line.

uint32_t _Line(byte Fx, Fy, Tx, Ty)
Combines four bytes into uint32_t (x from, y from, x to, y to)

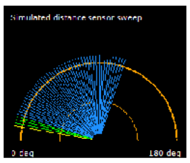
uint16_t _VPoint(byte x, byte y)
uint32_t _VLine(byte Fx, Fy, Tx, Ty)
Same as _Point and _Line but transform y values from value (0-255) to coordinate (0-220).

Helper function Float string
char * _FString(floatNumber, length, decimals);

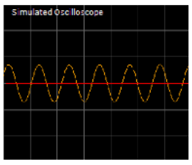
Panel Delay function
bool Panel.Delay(int16_t milliseconds, bool receive)
Allows to check for incoming messages during delay. If receive is true. Panel receive is called. If an incoming message was detected true is returned.

Graph Panel

Supports simple graphical display functions (rolling graph, static graph, free draw) including 4 extra buttons and 5 labels with color bars to associate with a graph.



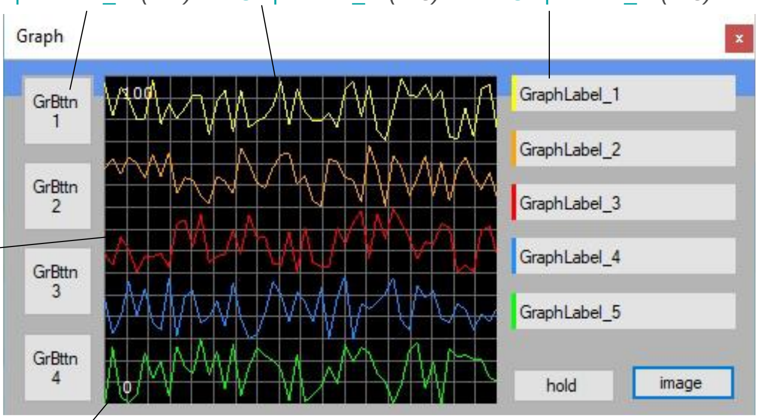
Graph Draw



Static Graph

GraphGrid
GraphDrawLine
GraphDrawPixel
GraphText
GraphClick
GraphRightClick
GraphDoubleClick

GraphButton_'x' *(1..4)* GraphValue_'x' *(1..5)* GraphLabel_'x' *(1..5)*



RolingGraph

GraphCaption_'x' *(1..2)*

hold / resume: discards incoming mesages
image: saves graph pane content to file

Graph channels/events

Graph send

bool	win. visible/invisible
string	$CLEAR

GraphGrid send

int16	vert. gridcount

GraphDrawLine send

void	Line start
uint16[2]	point 2 x byte (x,y)
uint32[2]	Line 4 x byte (Fx,Fy,Tx,Ty)
color[1]	line color
width[1]	line width string

GraphDrawPixel send

color[1]	pixel color
uint16[2]	point 2 x byte (x,y)

GraphCaption_'x' (1..2) send

any	Caption text

GraphText send

color[1]	text color
uint16[2]	point 2 x byte (x,y)
string	text

GraphValue_'x' (1..5) send

byte	point 2 x byte (x,y)
color[1]	Graph color
width[1]	line width string
type[1]	rolling/static
$CLEAR	clear sent values

GraphValueCount_'x' (1..5) send

int16	hor. value count

[1]See: *Special strings*

[2] Helper functions:
uint16_t _Point(byte x, byte y)
Uint32_t _Line(byte Fx, Fy, Tx, Ty)

Graph Panel 255(x) X 220(y)
Actual 263(x) for GraphValue

GraphLabel_'x' (1..5) send

bool	visible/invisible
any	label text
color[1]	color bar color*

* $OFF (color bar invisible)

GraphButton_'x' (1..4) send

any	button text
color[1]	button color
size[1]	text size

receive

void	on button click

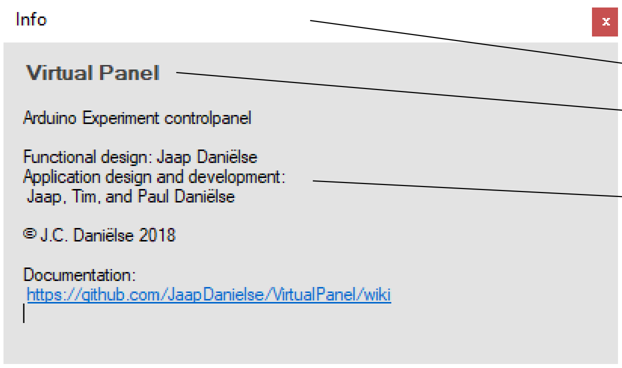
GraphClick receive
GraphRightClick receive
GraphDoubleClick* receive

uint16**	point 2 x byte (x,y)

* occurs together with GraphClick
**uint 2 x byte (X,Y)
(same as DrawPoint and DrawLine)

Info Panel

Application dependent help panel.



Virtual Panel

Arduino Experiment controlpanel

Functional design: Jaap Daniëlse
Application design and development:
Jaap, Tim, and Paul Daniëlse

© J.C. Daniëlse 2018

Documentation:
https://github.com/JaapDanielse/VirtualPanel/wiki

Info
InfoTitle
InfoText

Info channels/ events

Info send

bool	win. visible/invisible
string	$CLEAR

InfoTitle send

any*	title text

*Also clears InfoText

InfoText send

string*	Info text
$CLEAR	Clears info text

* max 60 char per send.
Can be repeated for larger text

Miscellaneous

Sendf() / Printf formatting

Limited list.

%[flags][width][length]specifier

specifiers

%d	signed decimal
%ld	unsigned int32
%u	unsigned decimal
%o	unsigned octal
%x	unsigned hex
%c	character
%s	string

flags

-	left justify
+	force sign
0	pad zero's

Examples:

Panel.sendf (Display_1, "Test %d", 10) // output: Test 10

Panel.sendf(Display_1, "Test %03d", 10) // output: Test 010

Panel.sendf(Display_1, "Test %+d", 10) // output: Test +10

sendf() float

Float not supported on AVR (Uno, Nano, Mega …)

Use _FString() helper function.

char* _FString(floatNumber, length, decimals); again with Panel.sendf using "%s"

Example:

Panel.sendf(Display_1, "Value %s" , _FString(FloatValue, 5, 2);

Prints FloatValue using 5 chars, 3 of which are a '.' and 2 decimals.

F() Macro

In both send() and sendf() the F() macro for strings is allowed. This will force the string to be placed in program memory. (*not Due*)

Example:

Panel.sendf
 (Display_1, F("Value %d"), 10);

Menu

Drop down from main panel.

Monitor	open/close monitor window*
Graph	open/close Graph window*
Message Log	open/close Msg.Log window
Reset Arduino	reset Arduino (not all processor types)
Info	open/close Info window *

* Can also be opened using channel.

Panel Variables

Event data received

Panel.vpr_void[4]	void
Panel.vpr_bool[5]	bool
Panel.vpr_string[3,5]	char*
Panel.vpr_byte[5]	byte
Panel.vpr_int[1,5]	int16_t
Panel.vpr_uint[2,5]	unint16_t
Panel.vpr_long[5]	int32_t
Panel.vpr_ulong[5]	unit32_t
Panel.vpr_float[5]	float32_t

[1] Slider_'x' *(value)*

[2] GraphClick, GraphRightClick, GraphDoubleClick *(point)*

[3] Max 35 char.

[4] Button_'x' *(click)*, PanelInput_'x', MonitorInput_'x' *(discard)*

[5] PanelInput_'x', MonitorInput_'x' *(value)*

Data type received

Panel.vpr_type	vpr_type

Data type names

Received in Panel.vpr_type

vp_type::vp_void	void
vp_type::vp_boolean	bool
vp_type::vp_string	char*
vp_type::vp_byte	byte
vp_type::vp_int	int16
vp_type::vp_uint	uint16
vp_type::vp_long	int32
vp_type::vp_ulong	uint32
vp_type::vp_float	float

See input snippet below

Code snippets

Button

```
Panel.send(Button_1, "on\noff");  //init
…
case Button_1:  // Button_1 case in event switch
   // Button_1 code
 break;
```

Slider

```
Panel.send(Slider_1, "level");  //set label
Panel.send(MaxSlider_1, 255);  //set max value
Panel.send(Slider_1, 127);  //set (initial) value
…
case Slider_1: // Slider_1 case in event switch
   MySliderValue = Panel.vpr_int;    // copy value
   // Slider_1 code
 break;
```

Input

```
case Display_1: // Display_1 double clicked
 Panel.send(PanelInputLabel_1, "Inp. value:"); //set label
 Panel.send(MinPanelInput_1, 0);  //set min. value
 Panel.send(MaxPanelInput_1, 100);  //set max. value
 Panel.send(PanelInput_1, 42);  //set current value
break;

case PanelInput_1: //PanelInput_1 case in event switch
  if (Panel.vpr_type != vpr_type::vp_void)  // check not discard
    MyInputValue = Panel.vpr_int;    // copy value
  // PanelInput_1 code
break;
```

Graph

```
Panel.send(GraphGrid, 10);  //set grid num. of vert sections
Panel.send(GraphValueCount_1, 100);  //set num. of value
Panel.send(GraphValue_1, "$RED");  //set color red
…
Panel.send(GraphValue_1,  Value);  //send value (def. rolling)
```