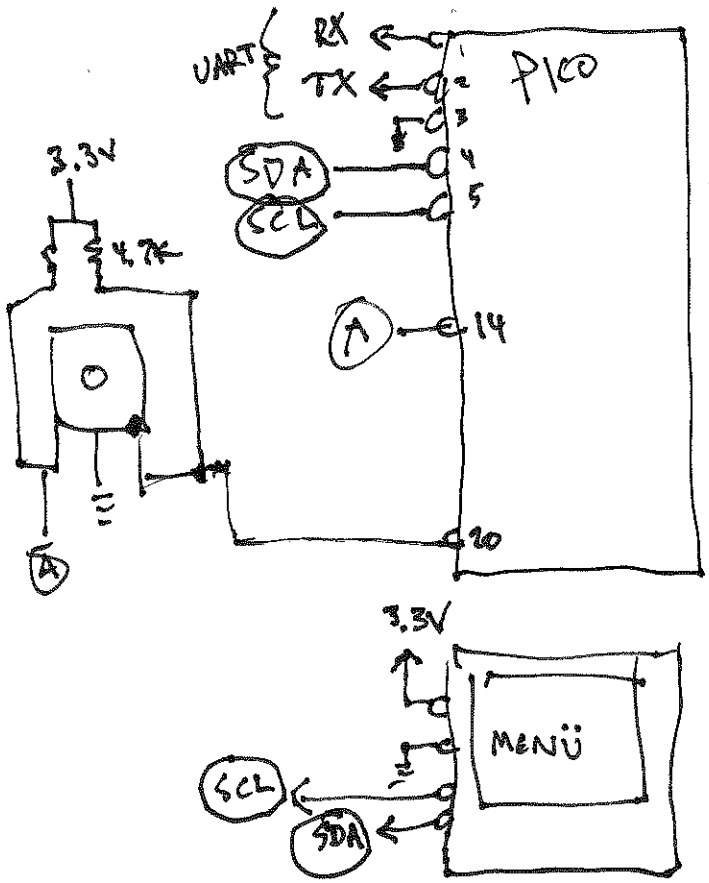
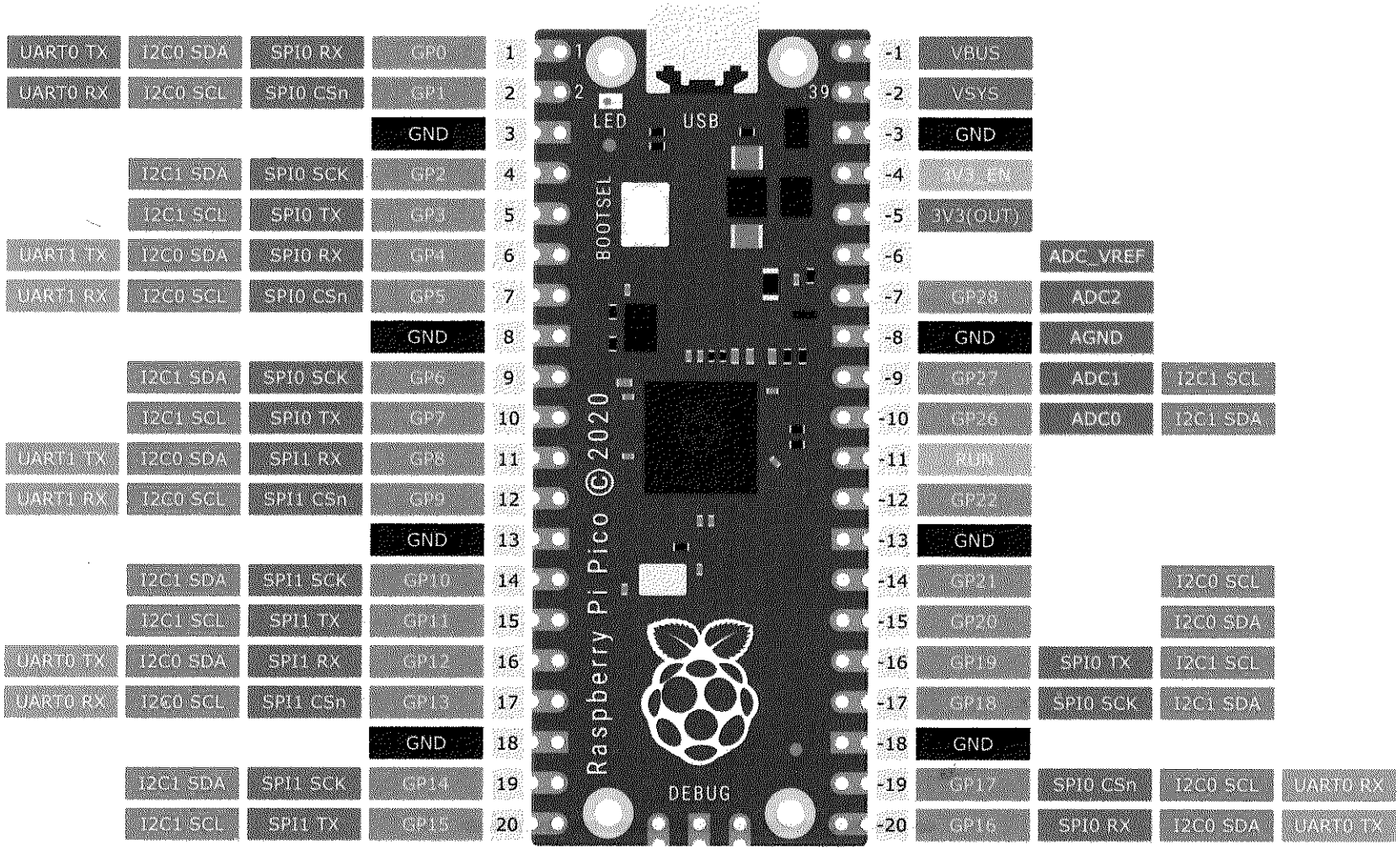


SSD1306 | Encoder Wiring


UART0 TX
I2C0 SDA
SPI0 RX
GP0
1
UART0 RX
I2C0 SCL
SPI0 CSn
GP1
2
GND
3
I2C1 SDA
SPI0 SCK
GP2
4
I2C1 SCL
SPI0 TX
GP3
5
UART1 TX
I2C0 SDA
SPI0 RX
GP4
6
UART1 RX
I2C0 SCL
SPI0 CSn
GP5
7
GND
8
I2C1 SDA
SPI0 SCK
GP6
9
I2C1 SCL
SPI0 TX
GP7
10
UART1 TX
I2C0 SDA
SPI1 RX
GP8
11
UART1 RX
I2C0 SCL
SPI1 CSn
GP9
12
GND
13
I2C1 SDA
SPI1 SCK
GP10
14
I2C1 SCL
SPI1 TX
GP11
15
UART0 TX
I2C0 SDA
SPI1 RX
GP12
16
UART0 RX
I2C0 SCL
SPI1 CSn
GP13
17
GND
18
I2C1 SDA
SPI1 SCK
GP14
19
I2C1 SCL
SPI1 TX
GP15
20
1
2
39
LED
USB
BOOTSEL
Raspberry Pi Pico ©2020
DEBUG
-1
VBUS
-2
VSYS
-3
GND
-4
3V3_EN
-5
3V3(OUT)
-6
ADC_VREF
-7
GP28
ADC2
-8
GND
AGND
-9
GP27
ADC1
I2C1 SCL
-10
GP26
ADC0
I2C1 SDA
-11
RUN
-12
GP22
-13
GND
-14
GP21
I2C0 SCL
-15
GP20
I2C0 SDA
-16
GP19
SPI0 TX
I2C1 SCL
-17
GP18
SPI0 SCK
I2C1 SDA
-18
GND
-19
GP17
SPI0 CSn
I2C0 SCL
UART0 RX
-20
GP16
SPI0 RX
I2C0 SDA
UART0 TX

daschr / **pico-ssd1306** Public

Code Issues 2 Pull requests Actions Projects Security Insights

main ▾ ...

pico-ssd1306 / ssd1306.h

daschr ssd13606_draw_empty_square -> ssd1306_draw_empty_square ✓ History

1 contributor

254 lines (207 sloc) | 7.04 KB ...

```
1  /*
2  MIT License
3
4  Copyright (c) 2021 David Schramm
5
6  Permission is hereby granted, free of charge, to any person obtaining a copy
7  of this software and associated documentation files (the "Software"), to deal
8  in the Software without restriction, including without limitation the rights
9  to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
10 copies of the Software, and to permit persons to whom the Software is
11 furnished to do so, subject to the following conditions:
12
13 The above copyright notice and this permission notice shall be included in all
14 copies or substantial portions of the Software.
15
16 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
17 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
18 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
19 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
20 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
21 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
22 SOFTWARE.
23 */
24
25 /**
26 * @file ssd1306.h
27 *
28 * simple driver for ssd1306 displays
29 */
```

```
30
31  #ifndef _inc_ssd1306
32  #define _inc_ssd1306
33  #include <pico/stdlib.h>
34  #include <hardware/i2c.h>
35
36  /**
37  *       @brief defines commands used in ssd1306
38  */
39  typedef enum {
40      SET_CONTRAST = 0x81,
41      SET_ENTIRE_ON = 0xA4,
42      SET_NORM_INV = 0xA6,
43      SET_DISP = 0xAE,
44      SET_MEM_ADDR = 0x20,
45      SET_COL_ADDR = 0x21,
46      SET_PAGE_ADDR = 0x22,
47      SET_DISP_START_LINE = 0x40,
48      SET_SEG_REMAP = 0xA0,
49      SET_MUX_RATIO = 0xA8,
50      SET_COM_OUT_DIR = 0xC0,
51      SET_DISP_OFFSET = 0xD3,
52      SET_COM_PIN_CFG = 0xDA,
53      SET_DISP_CLK_DIV = 0xD5,
54      SET_PRECHARGE = 0xD9,
55      SET_VCOM_DESEL = 0xDB,
56      SET_CHARGE_PUMP = 0x8D
57  } ssd1306_command_t;
58
59  /**
60  *       @brief holds the configuration
61  */
62  typedef struct {
63      uint8_t width;              /**< width of display */
64      uint8_t height;     /**< height of display */
65      uint8_t pages;              /**< stores pages of display (calculated on initialization*/
66      uint8_t address;    /**< i2c address of display*/
67      i2c_inst_t *i2c_i;  /**< i2c connection instance */
68      bool external_vcc;  /**< whether display uses external vcc */
69      uint8_t *buffer;    /**< display buffer */
70      size_t bufsize;             /**< buffer size */
71  } ssd1306_t;
72
73  /**
74  *       @brief initialize display
75  *
76  *       @param[in] p : pointer to instance of ssd1306_t
77  *       @param[in] width : width of display
78  *       @param[in] height : heigth of display
```

```
 79  *       @param[in] address : i2c address of display
 80  *       @param[in] i2c_instance : instance of i2c connection
 81  *
 82  *       @return bool.
 83  *       @retval true for Success
 84  *       @retval false if initialization failed
 85  */
 86  bool ssd1306_init(ssd1306_t *p, uint16_t width, uint16_t height, uint8_t address, i2c_inst_t *i2c_
 87
 88  /**
 89  *       @brief deinitialize display
 90  *
 91  *       @param[in] p : instance of display
 92  *
 93  */
 94  void ssd1306_deinit(ssd1306_t *p);
 95
 96  /**
 97  *       @brief turn off display
 98  *
 99  *       @param[in] p : instance of display
100  *
101  */
102  void ssd1306_poweroff(ssd1306_t *p);
103
104  /**
105          @brief turn on display
106
107          @param[in] p : instance of display
108
109  */
110  void ssd1306_poweron(ssd1306_t *p);
111
112  /**
113          @brief set contrast of display
114
115          @param[in] p : instance of display
116          @param[in] val : contrast
117
118  */
119  void ssd1306_contrast(ssd1306_t *p, uint8_t val);
120
121  /**
122          @brief set invert display
123
124          @param[in] p : instance of display
125          @param[in] inv : inv==0: disable inverting, inv!=0: invert
126
127  */
```

```
128 void ssd1306_invert(ssd1306_t *p, uint8_t inv);
129
130 /**
131         @brief display buffer, should be called on change
132
133         @param[in] p : instance of display
134
135 */
136 void ssd1306_show(ssd1306_t *p);
137
138 /**
139         @brief clear display buffer
140
141         @param[in] p : instance of display
142
143 */
144 void ssd1306_clear(ssd1306_t *p);
145
146 /**
147         @brief draw pixel on buffer
148
149         @param[in] p : instance of display
150         @param[in] x : x position
151         @param[in] y : y position
152 */
153 void ssd1306_draw_pixel(ssd1306_t *p, uint32_t x, uint32_t y);
154
155 /**
156         @brief draw pixel on buffer
157
158         @param[in] p : instance of display
159         @param[in] x1 : x position of starting point
160         @param[in] y1 : y position of starting point
161         @param[in] x2 : x position of end point
162         @param[in] y2 : y position of end point
163 */
164 void ssd1306_draw_line(ssd1306_t *p, int32_t x1, int32_t y1, int32_t x2, int32_t y2);
165
166 /**
167         @brief draw filled square at given position with given size
168
169         @param[in] p : instance of display
170         @param[in] x : x position of starting point
171         @param[in] y : y position of starting point
172         @param[in] width : width of square
173         @param[in] height : height of square
174 */
175 void ssd1306_draw_square(ssd1306_t *p, uint32_t x, uint32_t y, uint32_t width, uint32_t height);
176
```

```
177 /**
178         @brief draw empty square at given position with given size
179
180         @param[in] p : instance of display
181         @param[in] x : x position of starting point
182         @param[in] y : y position of starting point
183         @param[in] width : width of square
184         @param[in] height : height of square
185 */
186 void ssd1306_draw_empty_square(ssd1306_t *p, uint32_t x, uint32_t y, uint32_t width, uint32_t heig
187
188 /**
189         @brief draw monochrome bitmap with offset
190
191         @param[in] p : instance of display
192         @param[in] data : image data (whole file)
193         @param[in] size : size of image data in bytes
194         @param[in] x_offset : offset of horizontal coordinate
195         @param[in] y_offset : offset of vertical coordinate
196 */
197 void ssd1306_bmp_show_image_with_offset(ssd1306_t *p, const uint8_t *data, const long size, uint32
198
199 /**
200         @brief draw monochrome bitmap
201
202         @param[in] p : instance of display
203         @param[in] data : image data (whole file)
204         @param[in] size : size of image data in bytes
205 */
206 void ssd1306_bmp_show_image(ssd1306_t *p, const uint8_t *data, const long size);
207
208 /**
209         @brief draw char with given font
210
211         @param[in] p : instance of display
212         @param[in] x : x starting position of char
213         @param[in] y : y starting position of char
214         @param[in] scale : scale font to n times of original size (default should be 1)
215         @param[in] font : pointer to font
216         @param[in] c : character to draw
217 */
218 void ssd1306_draw_char_with_font(ssd1306_t *p, uint32_t x, uint32_t y, uint32_t scale, const uint8
219
220 /**
221         @brief draw char with builtin font
222
223         @param[in] p : instance of display
224         @param[in] x : x starting position of char
225         @param[in] y : y starting position of char
```

```
225         @param[in] scale : scale font to n times of original size (default should be 1)
227         @param[in] c : character to draw
228 */
229 void ssd1306_draw_char(ssd1306_t *p, uint32_t x, uint32_t y, uint32_t scale, char c);
230
231 /**
232         @brief draw string with given font
233
234         @param[in] p : instance of display
235         @param[in] x : x starting position of text
236         @param[in] y : y starting position of text
237         @param[in] scale : scale font to n times of original size (default should be 1)
238         @param[in] font : pointer to font
239         @param[in] s : text to draw
240 */
241 void ssd1306_draw_string_with_font(ssd1306_t *p, uint32_t x, uint32_t y, uint32_t scale, const uin
242
243 /**
244         @brief draw string with builtin font
245
246         @param[in] p : instance of display
247         @param[in] x : x starting position of text
248         @param[in] y : y starting position of text
249         @param[in] scale : scale font to n times of original size (default should be 1)
250         @param[in] s : text to draw
251 */
252 void ssd1306_draw_string(ssd1306_t *p, uint32_t x, uint32_t y, uint32_t scale, const char *s);
253
254 #endif
```