Plunger firmware

V0.1

Generated by Doxygen 1.12.0

Chapter 1

Plunger firmware

Main file containing the setup and main loop functions.

Main file containing the setup and main loop functions.

1.1 Introduction

This project handles user interface interactions using a rotary encoder and buttons, allowing for navigation through menus, adjustment of settings, and controlling a display. It implements a multitasking approach using FreeRTOS to manage various tasks concurrently.

1.2 dependencies

Dependencies: The libraries needed for this to compile are:

- Arduino.h
- FreeRTOS
- BfButton
- TFT_eSPI
- EthernetENC
- SPI
- ThreeWire
- RtcDS1302
- AccelStepper

1.3 Author

Written by GranaSAT Students:

Author

Juan Alberto Serrano Redondo `juan@ugr.es`
Prof. Andrés Roldán Aranda `amroldan@ugr.es`

1.4 License

BSD license, all text here must be included in any redistribution.

1.5 Flow chart of the project

1.6 HISTORY

V00 (07/2024):

- Implemented rotary encoder input handling.
- Integrated button press detection (single, double, long presses).
- Configured display updates based on menu navigation.
- Developed tasks using FreeRTOS for efficient multitasking. V00.1 (08/2024):
- Added Ethernet communication features for remote monitoring.
- Implemented real-time clock (RTC) functionality for accurate time management.
- Enhanced display functionality to show system status and settings.
- Improved input handling to support simultaneous button and encoder interactions.
- Introduced settings menu for configuring system parameters.
- Added logging functionality for troubleshooting and performance analysis.
- Optimized task scheduling to reduce CPU load and improve responsiveness.
- Refined user interface design for better user experience on the TFT display.

V01 (10/2024):

- Implemented data visualization features for displaying sensor readings.

- Developed communication protocols for data transmission over Ethernet.
- Conducted extensive testing and debugging to ensure system reliability.
- Refined rotary encoder response for smoother menu navigation.
- Implemented power management features to reduce energy consumption.
- Expanded documentation and comments for better maintainability.

This file includes the global configurations and necessary libraries for the project. The setup function initializes all configurations, including RTOS (Real-Time Operating System) tasks, which manage the entire functionality of the system.

The main loop (`void loop()`) is intentionally left empty, as the RTOS tasks are responsible for the continuous execution of system tasks, following a multi-threading approach. The tasks are created and managed in the setup function.

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

File Documentation

3.1 button/button.cpp File Reference

Implementation file for button press handling functions.

```
#include "button.h"
#include <globals/globals.h>
#include <motor/motor.h>
#include <GUI/GUI.h>
```

Functions

- void singlePressHandler (BfButton ∗btn, BfButton::press_pattern_t pattern)

  Handles single button press events.
- void doublePressHandler (BfButton ∗btn, BfButton::press_pattern_t pattern)

  Handles double button press events.
- void longPressHandler (BfButton ∗btn, BfButton::press_pattern_t pattern)

  Handles long button press events.

3.1.1 Detailed Description

Implementation file for button press handling functions.

This file contains the implementation of functions that handle button press events, including single, double, and long presses. These functions allow users to navigate through menus, control the motor, adjust screen settings, and perform other actions.

3.1.2 Function Documentation

3.1.2.1 doublePressHandler()

```
void doublePressHandler (
            BfButton * btn,
            BfButton::press_pattern_t pattern)
```

Handles double button press events.

This function toggles the engineering mode when a double press is detected.

Parameters

btn	Pointer to the button object.
pattern	The press pattern detected (double press in this case).

3.1.2.2 longPressHandler()

```
void longPressHandler (
            BfButton * btn,
            BfButton::press_pattern_t pattern)
```

Handles long button press events.

This function toggles the screen on/off. If the screen is turned off, it reduces the brightness to zero and puts the display into sleep mode. When turned back on, the display is woken up and the brightness is restored to the previous level.

Parameters

btn	Pointer to the button object.
pattern	The press pattern detected (long press in this case).

3.1.2.3 singlePressHandler()

```
void singlePressHandler (
            BfButton * btn,
            BfButton::press_pattern_t pattern)
```

Handles single button press events.

This function manages the navigation between different menus (main menu, settings, motor control, etc.) based on the current state and selection. It sends update commands to the screen to reflect the changes in the UI.

Parameters

btn	Pointer to the button object.
pattern	The press pattern detected (single press in this case).

3.2 button/button.h File Reference

Header file for button press handling functions.

```
#include <libraries.h>
```

Functions

- void singlePressHandler (BfButton ∗btn, BfButton::press_pattern_t pattern)

  Handles single button press events.
- void doublePressHandler (BfButton ∗btn, BfButton::press_pattern_t pattern)

  Handles double button press events.
- void longPressHandler (BfButton ∗btn, BfButton::press_pattern_t pattern)

  Handles long button press events.

3.2.1 Detailed Description

Header file for button press handling functions.

This file contains the declarations of the button press handling functions, including single, double, and long press actions for navigating the menu and controlling various settings.

3.2.2 Function Documentation

3.2.2.1 doublePressHandler()

```
void doublePressHandler (
             BfButton * btn,
             BfButton::press_pattern_t pattern)
```

Handles double button press events.

This function is called when a double press is detected. It toggles the engineering mode or any other logic that needs to be implemented for a double press event.

Parameters

btn	Pointer to the button object.
pattern	The press pattern detected (double press in this case).

This function toggles the engineering mode when a double press is detected.

Parameters

btn	Pointer to the button object.
pattern	The press pattern detected (double press in this case).

3.2.2.2 longPressHandler()

```
void longPressHandler (
             BfButton * btn,
             BfButton::press_pattern_t pattern)
```

Handles long button press events.

This function is called when a long press is detected. It toggles the screen on/off, adjusts the brightness, and manages the sleep mode of the display.

Parameters

btn	Pointer to the button object.
pattern	The press pattern detected (long press in this case).

This function toggles the screen on/off. If the screen is turned off, it reduces the brightness to zero and puts the display into sleep mode. When turned back on, the display is woken up and the brightness is restored to the previous level.

Parameters

btn	Pointer to the button object.
pattern	The press pattern detected (long press in this case).

3.2.2.3 singlePressHandler()

```
void singlePressHandler (
            BfButton * btn,
            BfButton::press_pattern_t pattern)
```

Handles single button press events.

This function is called when a single press is detected. It manages the navigation through different menus and performs specific actions based on the current menu and selection.

Parameters

btn	Pointer to the button object.
pattern	The press pattern detected (single press in this case).

This function manages the navigation between different menus (main menu, settings, motor control, etc.) based on the current state and selection. It sends update commands to the screen to reflect the changes in the UI.

Parameters

btn	Pointer to the button object.
pattern	The press pattern detected (single press in this case).

3.3 button.h

Go to the documentation of this file.

```

#ifndef BUTTON_H
#define BUTTON_H
#include <libraries.h>

void singlePressHandler(BfButton *btn, BfButton::press_pattern_t pattern);

void doublePressHandler(BfButton *btn, BfButton::press_pattern_t pattern);

void longPressHandler(BfButton *btn, BfButton::press_pattern_t pattern);

#endif // BUTTON_H
```

3.4 arimo12.h

```
// Created by https://oleddisplay.squix.ch/ Consider a donation
// In case of problems make sure that you are using the font file with the correct version!
const uint8_t Arimo_Regular_12Bitmaps[] PROGMEM = {

    // Bitmap Data:
    0x00, // ' '
    0xAA,0xA8,0x80, // '!'
    0x52,0x94, // '"'
    0x24,0x24,0xFE,0x24,0x28,0x48,0xFE,0x48,0x48, // '#'
    0x10,0x7C,0x54,0x50,0x70,0x3C,0x16,0x12,0xD6,0x7C,0x10, // '$'
    0x71,0x09,0x10,0x92,0x09,0x20,0x75,0xC0,0x92,0x09,0x21,0x12,0x11,0xC0, // '%'
    0x18,0x12,0x09,0x07,0x07,0x22,0x92,0x30,0x88,0x7B,0x00, // '&'
    0xA8, // '''
    0x44,0x88,0x88,0x88,0x84,0x60, // '('
    0x44,0x22,0x22,0x22,0x24,0x40, // ')'
    0x23,0xE2,0x14, // '*'
    0x20,0x8F,0x88,0x20, // '+'
    0xA8, // ','
    0xC0, // '-'
    0x80, // '.'
    0x22,0x44,0x44,0x48,0x80, // '/'
    0x38,0x44,0x44,0x44,0xC6,0x44,0x44,0x44,0x38, // '0'
    0x62,0x82,0x08,0x20,0x82,0x08,0xF8, // '1'
    0x72,0x20,0x82,0x10,0x84,0x20,0xF8, // '2'
    0x38,0x88,0x10,0x21,0x80,0x81,0x22,0x38, // '3'
    0x08,0x18,0x18,0x28,0x48,0x48,0xFE,0x08,0x08, // '4'
    0x7C,0x81,0x03,0xC0,0x40,0x81,0x22,0x78, // '5'
    0x72,0x28,0x3C,0x8A,0x28,0xA2,0x70, // '6'
    0xF8,0x21,0x04,0x20,0x82,0x10,0x40, // '7'
    0x72,0x28,0xA2,0x72,0x28,0xA2,0x70, // '8'
    0x72,0x28,0xA2,0x89,0xE0,0xA2,0x70, // '9'
    0x80,0x08, // ':'
    0x80,0x0A,0x80, // ';'
    0x09,0xC8,0x1C,0x08, // '<'
    0xF8,0x00,0x3E, // '='
    0x81,0xC0,0x9C,0x80, // '>'
    0x71,0x10,0x30,0x41,0x86,0x08,0x00,0x20, // '?'
    0x1F,0x04,0x11,0x35,0x49,0xAA,0x25,0x44,0xA8,0x94,0xFC,0x80,0x08,0x20,0xF8,0x00, // '@'
    0x10,0x28,0x28,0x28,0x44,0x7C,0x44,0xC6,0x82, // 'A'
    0xF9,0x0A,0x14,0x6F,0x90,0xA1,0x42,0xF8, // 'B'
    0x3C,0x61,0x20,0x10,0x08,0x04,0x02,0x00,0x84,0x3C,0x00, // 'C'
    0xF8,0x86,0x82,0x82,0x82,0x82,0x82,0x84,0xF8, // 'D'
    0xFD,0x02,0x04,0x0F,0xD0,0x20,0x40,0xFC, // 'E'
    0xFD,0x02,0x04,0x0F,0x90,0x20,0x40,0x80, // 'F'
    0x7C,0xC6,0x80,0x80,0x8E,0x82,0x82,0xC6,0x3C, // 'G'
    0x82,0x82,0x82,0x82,0xFE,0x82,0x82,0x82,0x82, // 'H'
    0xAA,0xAA,0x80, // 'I'
    0x38,0x20,0x82,0x08,0x20,0xA2,0x70, // 'J'
    0x84,0x88,0x90,0xA0,0xE0,0x90,0x88,0x8C,0x84, // 'K'
    0x81,0x02,0x04,0x08,0x10,0x20,0x40,0xFC, // 'L'
    0xC2,0xC6,0xC6,0xC6,0xAA,0xAA,0xAA,0x92,0x92, // 'M'
    0xC2,0xC2,0xA2,0xA2,0x92,0x8A,0x8A,0x86,0x86, // 'N'
    0x7C,0xC6,0x82,0x82,0x82,0x82,0x82,0xC6,0x7C, // 'O'
    0xF9,0x0A,0x14,0x28,0x5F,0x20,0x40,0x80, // 'P'
    0x7C,0xC6,0x82,0x82,0x82,0x82,0x82,0xC6,0x7C,0x18,0x0C, // 'Q'
    0xFC,0x82,0x82,0x82,0xFC,0x88,0x84,0x86,0x82, // 'R'
    0x79,0x0A,0x07,0x01,0xC0,0x81,0x42,0x78, // 'S'
    0xFE,0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x10, // 'T'
    0x82,0x82,0x82,0x82,0x82,0x82,0x82,0xC6,0x7C, // 'U'
    0x82,0x44,0x44,0x44,0x6C,0x28,0x28,0x28,0x10, // 'V'
    0x84,0x2C,0x46,0x4A,0x44,0xA4,0x4A,0x46,0xAC,0x31,0x83,0x18,0x31,0x80, // 'W'
    0x44,0x44,0x28,0x38,0x10,0x28,0x28,0x44,0x44, // 'X'
    0x44,0x44,0x28,0x28,0x10,0x10,0x10,0x10,0x10, // 'Y'
    0x7C,0x04,0x08,0x18,0x10,0x20,0x60,0x40,0xFE, // 'Z'
    0xD2,0x49,0x24,0x93,0x00, // '['
    0x88,0x44,0x44,0x42,0x20, // '\'
    0xC9,0x24,0x92,0x4B,0x00, // ']'
    0x21,0x45,0x22, // '^'
    0xFE, // '_'
    0x88, // '`'
    0x71,0x10,0x27,0xC8,0x93,0x3B,0x00, // 'a'
    0x82,0x0F,0xA2,0x8A,0x28,0xA2,0xF8, // 'b'
    0x72,0x68,0x20,0x82,0x67,0x00, // 'c'
    0x08,0x2F,0xA2,0x8A,0x28,0xA2,0xF8, // 'd'
    0x72,0x28,0xBE,0x82,0x27,0x00, // 'e'
    0x64,0xE4,0x44,0x44,0x40, // 'f'
    0xFA,0x28,0xA2,0x8A,0x2F,0x82,0xF0, // 'g'
    0x82,0x0F,0xA2,0x8A,0x28,0xA2,0x88, // 'h'
    0x8A,0xAA,0x80, // 'i'
    0x41,0x24,0x92,0x4B,0x00, // 'j'
    0x82,0x09,0x2C,0xA3,0x8A,0x24,0x98, // 'k'
    0xAA,0xAA,0x80, // 'l'
    0xFF,0xA2,0x28,0x8A,0x22,0x88,0xA2,0x28,0x88, // 'm'
    0xFA,0x28,0xA2,0x8A,0x28,0x80, // 'n'
    0x72,0x28,0xA2,0x8A,0x27,0x00, // 'o'
    0xFA,0x28,0xA2,0x8A,0x2F,0xA0,0x80, // 'p'
    0xFA,0x28,0xA2,0x8A,0x2F,0x82,0x08, // 'q'
    0xE8,0x88,0x88,0x80, // 'r'
    0x7C,0x44,0x40,0x3C,0x04,0x84,0x7C, // 's'
    0x44,0xE4,0x44,0x44,0x60, // 't'
    0x8A,0x28,0xA2,0x8A,0x2F,0x80, // 'u'
    0x8A,0x25,0x14,0x51,0x42,0x00, // 'v'
    0x88,0xA5,0x25,0x51,0x54,0x55,0x15,0x42,0x20, // 'w'
    0xD9,0x45,0x08,0x51,0x49,0x80, // 'x'
    0x8B,0x25,0x14,0x50,0x82,0x08,0xC0, // 'y'
    0x70,0x43,0x08,0x61,0x0F,0x80, // 'z'
    0x32,0x10,0x84,0x61,0x08,0x42,0x0C, // '{'
    0xAA,0xAA,0xAA, // '|'
    0xC1,0x08,0x42,0x18,0x84,0x23,0x30 // '}'
};
const GFXglyph Arimo_Regular_12Glyphs[] PROGMEM = {
// bitmapOffset, width, height, xAdvance, xOffset, yOffset
      {     0,   2,   1,   4,    0,   -1 }, // ' '
      {     1,   2,   9,   4,    1,   -9 }, // '!'
      {     4,   5,   3,   5,    0,   -9 }, // '"'
      {     6,   8,   9,   8,    0,   -9 }, // '#'
      {    15,   8,  11,   8,    0,  -10 }, // '$'
      {    26,  12,   9,  12,    0,   -9 }, // '%'
      {    40,   9,   9,   9,    0,   -9 }, // '&'
      {    51,   2,   3,   3,    1,   -9 }, // '''
      {    52,   4,  11,   5,    1,   -9 }, // '('
      {    58,   4,  11,   5,    0,   -9 }, // ')'
      {    64,   6,   4,   6,    0,   -9 }, // '*'
      {    67,   6,   5,   8,    1,   -7 }, // '+'
      {    71,   2,   3,   4,    1,   -1 }, // ','
      {    72,   3,   1,   5,    1,   -4 }, // '-'
      {    73,   2,   1,   4,    1,   -1 }, // '.'
      {    74,   4,   9,   4,    0,   -9 }, // '/'
      {    79,   8,   9,   8,    0,   -9 }, // '0'
      {    88,   6,   9,   8,    1,   -9 }, // '1'
      {    95,   6,   9,   8,    1,   -9 }, // '2'
      {   102,   7,   9,   8,    0,   -9 }, // '3'
      {   110,   8,   9,   8,    0,   -9 }, // '4'
      {   119,   7,   9,   8,    0,   -9 }, // '5'
      {   127,   6,   9,   8,    1,   -9 }, // '6'
      {   134,   6,   9,   8,    1,   -9 }, // '7'
      {   141,   6,   9,   8,    1,   -9 }, // '8'
      {   148,   6,   9,   8,    1,   -9 }, // '9'
      {   155,   2,   7,   4,    1,   -7 }, // ':'
      {   157,   2,   9,   4,    1,   -7 }, // ';'
      {   160,   6,   5,   8,    1,   -7 }, // '<'
      {   164,   6,   4,   8,    1,   -6 }, // '='
      {   167,   6,   5,   8,    1,   -7 }, // '>'
      {   171,   7,   9,   8,    1,   -9 }, // '?'
      {   179,  11,  11,  13,    1,   -9 }, // '@'
      {   195,   8,   9,   8,    0,   -9 }, // 'A'
      {   204,   7,   9,   9,    1,   -9 }, // 'B'
      {   212,   9,   9,  10,    1,   -9 }, // 'C'
      {   223,   8,   9,  10,    1,   -9 }, // 'D'
      {   232,   7,   9,   9,    1,   -9 }, // 'E'
      {   240,   7,   9,   8,    1,   -9 }, // 'F'
      {   248,   8,   9,  10,    1,   -9 }, // 'G'
      {   257,   8,   9,  10,    1,   -9 }, // 'H'
      {   266,   2,   9,   4,    1,   -9 }, // 'I'
      {   269,   6,   9,   7,    0,   -9 }, // 'J'
      {   276,   8,   9,   9,    1,   -9 }, // 'K'
      {   285,   7,   9,   8,    1,   -9 }, // 'L'
      {   293,   8,   9,  10,    1,   -9 }, // 'M'
      {   302,   8,   9,  10,    1,   -9 }, // 'N'
      {   311,   8,   9,  10,    1,   -9 }, // 'O'
      {   320,   7,   9,   9,    1,   -9 }, // 'P'
      {   328,   8,  11,  10,    1,   -9 }, // 'Q'
      {   339,   8,   9,  10,    1,   -9 }, // 'R'
      {   348,   7,   9,   9,    1,   -9 }, // 'S'
      {   356,   8,   9,   8,    0,   -9 }, // 'T'
      {   365,   8,   9,  10,    1,   -9 }, // 'U'
      {   374,   8,   9,   8,    0,   -9 }, // 'V'
      {   383,  12,   9,  12,    0,   -9 }, // 'W'
      {   397,   8,   9,   8,    0,   -9 }, // 'X'
      {   406,   8,   9,   8,    0,   -9 }, // 'Y'
      {   415,   8,   9,   8,    0,   -9 }, // 'Z'
      {   424,   3,  11,   4,    1,   -9 }, // '['
      {   429,   4,   9,   4,    0,   -9 }, // '\'
      {   434,   3,  11,   4,    0,   -9 }, // ']'
      {   439,   6,   4,   6,    0,   -9 }, // '^'
      {   442,   8,   1,   8,    0,    2 }, // '_'
      {   443,   3,   2,   5,    1,  -10 }, // '`'
      {   444,   7,   7,   8,    1,   -7 }, // 'a'
      {   451,   6,   9,   8,    1,   -9 }, // 'b'
      {   458,   6,   7,   7,    1,   -7 }, // 'c'
      {   464,   6,   9,   8,    1,   -9 }, // 'd'
      {   471,   6,   7,   8,    1,   -7 }, // 'e'
```

```
        {   477,   4,   9,   4,    0,   -9 }, // 'f'
        {   482,   6,   9,   8,    1,   -7 }, // 'g'
        {   489,   6,   9,   8,    1,   -9 }, // 'h'
        {   496,   2,   9,   4,    1,   -9 }, // 'i'
        {   499,   3,  11,   4,    0,   -9 }, // 'j'
        {   504,   6,   9,   7,    1,   -9 }, // 'k'
        {   511,   2,   9,   4,    1,   -9 }, // 'l'
        {   514,  10,   7,  12,    1,   -7 }, // 'm'
        {   523,   6,   7,   8,    1,   -7 }, // 'n'
        {   529,   6,   7,   8,    1,   -7 }, // 'o'
        {   535,   6,   9,   8,    1,   -7 }, // 'p'
        {   542,   6,   9,   8,    1,   -7 }, // 'q'
        {   549,   4,   7,   5,    1,   -7 }, // 'r'
        {   553,   8,   7,   8,    0,   -7 }, // 's'
        {   560,   4,   9,   4,    0,   -9 }, // 't'
        {   565,   6,   7,   8,    1,   -7 }, // 'u'
        {   571,   6,   7,   6,    0,   -7 }, // 'v'
        {   577,  10,   7,  10,    0,   -7 }, // 'w'
        {   586,   6,   7,   6,    0,   -7 }, // 'x'
        {   592,   6,   9,   6,    0,   -7 }, // 'y'
        {   599,   6,   7,   6,    0,   -7 }, // 'z'
        {   605,   5,  11,   5,    0,   -9 }, // '{'
        {   612,   2,  12,   4,    1,   -9 }, // '|'
        {   615,   5,  11,   5,    0,   -9 } // '}'
};
const GFXfont Arimo_Regular_12 PROGMEM = {
(uint8_t  *)Arimo_Regular_12Bitmaps,(GFXglyph *)Arimo_Regular_12Glyphs,0x20, 0x7E, 14};
```

3.5 arimo9.h

```
// Created by https://oleddisplay.squix.ch/ Consider a donation
// In case of problems make sure that you are using the font file with the correct version!
const uint8_t Arimo_Regular_9Bitmaps[] PROGMEM = {

    // Bitmap Data:
    0x00, // ' '
    0xAA,0x88, // '!'
    0xAA, // '"'
    0x01,0x4F,0x94,0xF9,0x40,0x00, // '#'
    0x21,0xCA,0x38,0x30,0xA2,0x9C,0x20, // '$'
    0x64,0x52,0x2A,0x0E,0xC1,0x51,0x28,0x98, // '%'
    0x30,0x91,0x43,0x49,0x93,0x1F,0x00, // '&'
    0xA0, // '''
    0x52,0x49,0x24,0x40, // '('
    0x89,0x24,0x92,0x80, // ')'
    0x45,0x99,0x20, // '*'
    0x20,0x8F,0x88,0x20, // '+'
    0xA0, // ','
    0xE0, // '-'
    0x80, // '.'
    0x22,0x44,0x48,0x80, // '/'
    0x72,0x48,0xA2,0x8A,0x47,0x00, // '0'
    0x46,0x10,0x84,0x23,0xC0, // '1'
    0x70,0x21,0x04,0x21,0x0F,0x80, // '2'
    0x72,0x21,0x0C,0x0A,0x27,0x00, // '3'
    0x10,0xC5,0x14,0x93,0xE1,0x00, // '4'
    0x71,0x07,0x02,0x0A,0x27,0x00, // '5'
    0x71,0x07,0x22,0x89,0x27,0x00, // '6'
    0xF8,0x41,0x08,0x20,0x82,0x00, // '7'
    0x72,0x25,0x1C,0x8A,0x27,0x00, // '8'
    0x72,0x48,0xA2,0x78,0x47,0x00, // '9'
    0x80,0x80, // ':'
    0x80,0xA0, // ';'
    0x19,0x88,0x18,0x18, // '<'
    0xF8,0x0F,0x80, // '='
    0xC0,0xC0,0x8C,0xC0, // '>'
    0x72,0x20,0x84,0x20,0x02,0x00, // '?'
    0x3C,0x42,0xBE,0xCA,0xCA,0xBC,0x80,0x78, // '@'
    0x30,0x60,0xA2,0x47,0x90,0xA1,0x00, // 'A'
    0xF2,0x49,0x3C,0x8A,0x2F,0x00, // 'B'
    0x3C,0x44,0x40,0x80,0x40,0x42,0x3C, // 'C'
    0xF1,0x12,0x14,0x28,0x91,0x3C,0x00, // 'D'
    0xF2,0x08,0x3C,0x82,0x0F,0x80, // 'E'
    0xFA,0x08,0x3C,0x82,0x08,0x00, // 'F'
    0x38,0x8A,0x04,0x68,0x48,0x8E,0x00, // 'G'
    0x8A,0x28,0xBE,0x8A,0x28,0x80, // 'H'
    0xAA,0xA8, // 'I'
    0x30,0x84,0x21,0x49,0xC0, // 'J'
    0x92,0x8C,0x30,0xA2,0x49,0x80, // 'K'
    0x84,0x21,0x08,0x43,0xC0, // 'L'
    0x8A,0x2D,0xB6,0xDA,0xAA,0x80, // 'M'
    0x8B,0x2C,0xAA,0x9A,0x68,0x80, // 'N'
    0x38,0x44,0x82,0x82,0x82,0x44,0x38, // 'O'
    0xF2,0x48,0xA4,0xE2,0x08,0x00, // 'P'
    0x38,0x44,0x82,0x82,0x82,0x44,0x38,0x10,0x0C, // 'Q'
    0xF9,0x12,0x27,0x89,0x11,0x22,0x00, // 'R'
    0x78,0x81,0x01,0xC0,0x50,0x9E,0x00, // 'S'
    0xF8,0x82,0x08,0x20,0x82,0x00, // 'T'
    0x8A,0x28,0xA2,0x8A,0x27,0x00, // 'U'
    0x84,0x91,0x22,0x43,0x06,0x0C,0x00, // 'V'
    0x88,0xA2,0x25,0x51,0x54,0x55,0x15,0x42,0x20, // 'W'
    0x51,0x42,0x08,0x51,0x48,0x80, // 'X'
    0x44,0x44,0x28,0x10,0x10,0x10,0x10, // 'Y'
    0x78,0x10,0x41,0x02,0x08,0x3F,0x00, // 'Z'
    0xD2,0x49,0x24,0xC0, // '['
    0x88,0x44,0x42,0x20, // '\'
    0xC9,0x24,0x92,0xC0, // ']'
    0x4A,0xA0, // '^'
    0xF8, // '_'
    0x48, // '`'
    0x70,0x47,0x24,0x68, // 'a'
    0x84,0x39,0x29,0x4B,0x80, // 'b'
    0x72,0x08,0x20,0x70, // 'c'
    0x10,0x9D,0x29,0x49,0xC0, // 'd'
    0x72,0x2F,0xA0,0x70, // 'e'
    0x32,0x3C,0x84,0x21,0x00, // 'f'
    0x74,0xA5,0x27,0x09,0xC0, // 'g'
    0x88,0xEA,0xAA,0xA0, // 'h'
    0x0A,0xA8, // 'i'
    0x0A,0xAA,0x80, // 'j'
    0x84,0x29,0x8C,0x52,0x40, // 'k'
    0xAA,0xA8, // 'l'
    0xED,0x2A,0x54,0xA9,0x40, // 'm'
    0xEA,0xAA,0xA0, // 'n'
    0x72,0x28,0xA2,0x70, // 'o'
    0xE4,0xA5,0x2E,0x42,0x00, // 'p'
    0x74,0xA5,0x27,0x08,0x40, // 'q'
    0xD2,0x48, // 'r'
    0x72,0x07,0x02,0x70, // 's'
    0x4E,0x44,0x46, // 't'
    0xAA,0xAA,0xE0, // 'u'
    0x84,0x91,0x21,0x83,0x00, // 'v'
    0xB5,0x6A,0xD2,0xC4,0x80, // 'w'
    0x51,0x42,0x14,0x88, // 'x'
    0x84,0x91,0x21,0x83,0x04,0x30,0x00, // 'y'
    0xF1,0x10,0x8F,0x00, // 'z'
    0x64,0x44,0x84,0x44,0x60, // '{'
    0xAA,0xAA,0x80, // '|'
    0xC4,0x44,0x24,0x44,0xC0 // '}'
};
const GFXglyph Arimo_Regular_9Glyphs[] PROGMEM = {
// bitmapOffset, width, height, xAdvance, xOffset, yOffset
      {     0,   2,   1,   4,    0,   -1 }, // ' '
      {     1,   2,   7,   4,    1,   -7 }, // '!'
      {     3,   4,   2,   4,    0,   -7 }, // '"'
      {     4,   6,   7,   6,    0,   -7 }, // '#'
      {    10,   6,   9,   6,    0,   -8 }, // '$'
      {    17,   9,   7,   9,    0,   -7 }, // '%'
      {    25,   7,   7,   7,    0,   -7 }, // '&'
      {    32,   2,   2,   3,    0,   -7 }, // '''
      {    33,   3,   9,   4,    1,   -7 }, // '('
      {    37,   3,   9,   4,    0,   -7 }, // ')'
      {    41,   5,   4,   5,    0,   -7 }, // '*'
      {    44,   6,   5,   6,    0,   -6 }, // '+'
      {    48,   2,   2,   4,    1,   -1 }, // ','
      {    49,   4,   1,   4,    0,   -3 }, // '-'
      {    50,   2,   1,   4,    1,   -1 }, // '.'
      {    51,   4,   7,   4,    0,   -7 }, // '/'
      {    55,   6,   7,   6,    0,   -7 }, // '0'
      {    61,   5,   7,   6,    1,   -7 }, // '1'
      {    66,   6,   7,   6,    0,   -7 }, // '2'
      {    72,   6,   7,   6,    0,   -7 }, // '3'
      {    78,   6,   7,   6,    0,   -7 }, // '4'
      {    84,   6,   7,   6,    0,   -7 }, // '5'
      {    90,   6,   7,   6,    0,   -7 }, // '6'
      {    96,   6,   7,   6,    0,   -7 }, // '7'
      {   102,   6,   7,   6,    0,   -7 }, // '8'
      {   108,   6,   7,   6,    0,   -7 }, // '9'
      {   114,   2,   5,   4,    1,   -5 }, // ':'
      {   116,   2,   6,   4,    1,   -5 }, // ';'
      {   118,   6,   5,   6,    0,   -6 }, // '<'
      {   122,   6,   3,   6,    0,   -5 }, // '='
      {   125,   6,   5,   6,    0,   -6 }, // '>'
      {   129,   6,   7,   6,    0,   -7 }, // '?'
      {   135,   8,   8,  10,    1,   -7 }, // '@'
      {   143,   7,   7,   7,    0,   -7 }, // 'A'
      {   150,   6,   7,   7,    1,   -7 }, // 'B'
      {   156,   8,   7,   8,    0,   -7 }, // 'C'
      {   163,   7,   7,   8,    1,   -7 }, // 'D'
      {   170,   6,   7,   7,    1,   -7 }, // 'E'
      {   176,   6,   7,   7,    1,   -7 }, // ’F’
      {   182,   7,   7,   8,    0,   -7 }, // ’G’
      {   189,   6,   7,   8,    1,   -7 }, // ’H’
      {   195,   2,   7,   4,    1,   -7 }, // ’I’
      {   197,   5,   7,   6,    0,   -7 }, // ’J’
      {   202,   6,   7,   7,    1,   -7 }, // ’K’
      {   208,   5,   7,   6,    1,   -7 }, // ’L’
      {   213,   6,   7,   8,    1,   -7 }, // ’M’
      {   219,   6,   7,   8,    1,   -7 }, // ’N’
      {   225,   8,   7,   8,    0,   -7 }, // ’O’
      {   232,   6,   7,   7,    1,   -7 }, // ’P’
      {   238,   8,   9,   8,    0,   -7 }, // ’Q’
      {   247,   7,   7,   8,    1,   -7 }, // ’R’
      {   254,   7,   7,   7,    0,   -7 }, // ’S’
      {   261,   6,   7,   6,    0,   -7 }, // ’T’
      {   267,   6,   7,   8,    1,   -7 }, // ’U’
      {   273,   7,   7,   7,    0,   -7 }, // ’V’
      {   280,  10,   7,  10,    0,   -7 }, // ’W’
      {   289,   6,   7,   6,    0,   -7 }, // ’X’
      {   295,   8,   7,   8,    0,   -7 }, // ’Y’
      {   302,   7,   7,   7,    0,   -7 }, // ’Z’
      {   309,   3,   9,   4,    1,   -7 }, // ’[’
      {   313,   4,   7,   4,    0,   -7 }, // ’\’
      {   317,   3,   9,   4,    0,   -7 }, // ’]’
      {   321,   4,   3,   4,    0,   -7 }, // ’^’
      {   323,   6,   1,   6,    0,    1 }, // ’_’
      {   324,   3,   2,   4,    0,   -8 }, // ’‘’
      {   325,   6,   5,   6,    0,   -5 }, // ’a’
      {   329,   5,   7,   6,    1,   -7 }, // ’b’
      {   334,   6,   5,   6,    0,   -5 }, // ’c’
      {   338,   5,   7,   6,    0,   -7 }, // ’d’
      {   343,   6,   5,   6,    0,   -5 }, // ’e’
      {   347,   5,   7,   5,    0,   -7 }, // ’f’
      {   352,   5,   7,   6,    0,   -5 }, // ’g’
      {   357,   4,   7,   6,    1,   -7 }, // ’h’
      {   361,   2,   7,   3,    1,   -7 }, // ’i’
      {   363,   2,   9,   3,    0,   -7 }, // ’j’
      {   366,   5,   7,   6,    1,   -7 }, // ’k’
      {   371,   2,   7,   3,    1,   -7 }, // ’l’
      {   373,   7,   5,   9,    1,   -5 }, // ’m’
      {   378,   4,   5,   6,    1,   -5 }, // ’n’
      {   381,   6,   5,   6,    0,   -5 }, // ’o’
      {   385,   5,   7,   6,    1,   -5 }, // ’p’
      {   390,   5,   7,   6,    0,   -5 }, // ’q’
      {   395,   3,   5,   4,    1,   -5 }, // ’r’
      {   397,   6,   5,   6,    0,   -5 }, // ’s’
      {   401,   4,   6,   4,    0,   -6 }, // ’t’
      {   404,   4,   5,   6,    1,   -5 }, // ’u’
      {   407,   7,   5,   7,    0,   -5 }, // ’v’
      {   412,   7,   5,   7,    0,   -5 }, // ’w’
      {   417,   6,   5,   6,    0,   -5 }, // ’x’
      {   421,   7,   7,   7,    0,   -5 }, // ’y’
      {   428,   5,   5,   5,    0,   -5 }, // ’z’
      {   432,   4,   9,   4,    0,   -7 }, // ’{’
      {   437,   2,   9,   4,    1,   -7 }, // ’|’
      {   440,   4,   9,   4,    0,   -7 } // ’}’
};
const GFXfont Arimo_Regular_9 PROGMEM = {
(uint8_t  *)Arimo_Regular_9Bitmaps,(GFXglyph *)Arimo_Regular_9Glyphs,0x20, 0x7E, 12};
```

3.6 custom_fonts.h

```
#include <custom_fonts/arimo9.h>
#include <custom_fonts/arimo12.h>
#include <custom_fonts/dialog9.h>
#include <custom_fonts/dialog12.h>
#include <custom_fonts/gothic9.h>
#include <custom_fonts/gothic12.h>
#include <custom_fonts/mono9.h>
#include <custom_fonts/mono12.h>
#include <custom_fonts/opensans9.h>
#include <custom_fonts/opensans12.h>
#include <custom_fonts/roboto9.h>
#include <custom_fonts/roboto12.h>
#include <custom_fonts/SansSerif9.h>
#include <custom_fonts/SansSerif12.h>
#include <custom_fonts/serif9.h>
#include <custom_fonts/serif12.h>




#define arimo9 &Arimo_Regular_9
```

```
#define arimo12 &Arimo_Regular_12
#define dialog9 &Dialog_plain_9
#define dialog12 &Dialog_plain_12
#define gothic9 &URW_Gothic_L_Book_9
#define gothic12 &URW_Gothic_L_Book_12
#define roboto9 &Roboto_Condensed_Light_9
#define roboto12 &Roboto_Condensed_Light_12
#define mono9 &Monospaced_plain_9
#define mono12 &Monospaced_plain_12
#define opensans9 &Open_Sans_Regular_9
#define opensans12 &Open_Sans_Regular_12
#define sansserif9 &SansSerif_plain_9
#define sansserif12 &SansSerif_plain_12
#define serif9 &Serif_plain_9
#define serif12 &Serif_plain_12
```

3.7 dialog12.h

```
// Created by https://oleddisplay.squix.ch/ Consider a donation
// In case of problems make sure that you are using the font file with the correct version!
const uint8_t Dialog_plain_12Bitmaps[] PROGMEM = {

    // Bitmap Data:
    0x00, // ' '
    0xAA,0xA2,0x80, // '!'
    0xAA,0xA0, // '"'
    0x12,0x0A,0x1F,0xC4,0x82,0x47,0xF0,0xA0,0x90, // '#'
    0x21,0xCA,0xA8,0xE0,0xE2,0xAA,0x70,0x82,0x00, // '$'
    0x61,0x12,0x42,0x48,0x4A,0x06,0xD8,0x14,0x84,0x90,0x92,0x21,0x80, // '%'
    0x30,0x24,0x10,0x0C,0x05,0x14,0x4A,0x19,0x88,0x7B,0x00, // '&'
    0xA8, // '''
    0x64,0x48,0x88,0x88,0x44,0x60, // '('
    0xC4,0x42,0x22,0x22,0x44,0xC0, // ')'
    0x22,0xA7,0x1C,0xA8,0x80, // '*'
    0x10,0x10,0x10,0xFE,0x10,0x10,0x10, // '+'
    0xA8, // ','
    0xE0, // '-'
    0xA0, // '.'
    0x10,0x88,0x42,0x21,0x08,0x84,0x00, // '/'
    0x78,0x92,0x14,0x28,0x50,0xA1,0x24,0x78, // '0'
    0xE0,0x82,0x08,0x20,0x82,0x08,0xF8, // '1'
    0x79,0x18,0x10,0x20,0x82,0x08,0x20,0xFC, // '2'
    0x79,0x08,0x10,0x23,0x80,0x81,0x42,0x78, // '3'
    0x18,0x30,0xA2,0x44,0x91,0x3F,0x04,0x08, // '4'
    0xF9,0x02,0x07,0xC0,0xC0,0x81,0x46,0x78, // '5'
    0x38,0x8A,0x05,0xCC,0xD0,0xA1,0x26,0x78, // '6'
    0xFC,0x08,0x20,0x41,0x02,0x08,0x10,0x40, // '7'
    0x79,0x0A,0x14,0x27,0x90,0xA1,0x42,0x78, // '8'
    0x79,0x92,0x14,0x2C,0xCE,0x81,0x44,0x70, // '9'
    0xA0,0xA0, // ':'
    0xA0,0xA8, // ';'
    0x03,0x0F,0x38,0x1C,0x01,0xE0,0x18, // '<'
    0xFF,0x00,0x3F,0xC0, // '='
    0xC0,0x3C,0x01,0xC0,0xE7,0x86,0x00, // '>'
    0x72,0x20,0x84,0x20,0x80,0x08,0x20, // '?'
    0x1F,0x02,0x0C,0x40,0x48,0xF2,0x91,0x29,0x12,0x91,0x48,0xF8,0x40,0x02,0x08,0x1F,0x00, // '@'
    0x18,0x0C,0x09,0x04,0x82,0x42,0x11,0xF8,0x84,0x81,0x00, // 'A'
    0xF9,0x0A,0x14,0x2F,0x90,0xA1,0x42,0xF8, // 'B'
    0x38,0x8A,0x04,0x08,0x10,0x20,0x22,0x38, // 'C'
    0xF8,0x84,0x82,0x82,0x82,0x82,0x82,0x84,0xF8, // 'D'
    0xFD,0x02,0x04,0x0F,0xD0,0x20,0x40,0xFC, // 'E'
    0xFA,0x08,0x20,0xFA,0x08,0x20,0x80, // 'F'
    0x3C,0x42,0x80,0x80,0x8E,0x82,0x82,0x42,0x3C, // 'G'
    0x82,0x82,0x82,0x82,0xFE,0x82,0x82,0x82,0x82, // 'H'
    0xAA,0xAA,0x80, // 'I'
    0x22,0x22,0x22,0x22,0x22,0xC0, // 'J'
    0x84,0x88,0x90,0xA0,0xC0,0xA0,0x90,0x88,0x84, // 'K'
    0x82,0x08,0x20,0x82,0x08,0x20,0xF8, // 'L'
    0x81,0x61,0xB0,0xD4,0xAA,0x54,0xCA,0x65,0x02,0x81,0x00, // 'M'
    0xC2,0xC2,0xA2,0xA2,0x92,0x8A,0x8A,0x86,0x86, // 'N'
    0x38,0x44,0x82,0x82,0x82,0x82,0x82,0x44,0x38, // 'O'
    0xF9,0x0A,0x14,0x2F,0x90,0x20,0x40,0x80, // 'P'
    0x38,0x44,0x82,0x82,0x82,0x82,0x82,0x44,0x38,0x08,0x04, // 'Q'
    0xF8,0x84,0x84,0x84,0xF8,0x88,0x84,0x84,0x82, // 'R'
    0x79,0x0A,0x04,0x07,0x80,0x81,0x42,0x78, // 'S'
    0xFE,0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x10, // 'T'
    0x82,0x82,0x82,0x82,0x82,0x82,0x82,0xC6,0x7C, // 'U'
    0x40,0x88,0x10,0x84,0x10,0x82,0x10,0x24,0x04,0x80,0x60,0x0C,0x00, // 'V'
    0x84,0x24,0x44,0x44,0x44,0xA4,0x2A,0x82,0xA8,0x2A,0x81,0x10,0x11,0x00, // 'W'
    0xC6,0x44,0x28,0x28,0x10,0x28,0x28,0x44,0x82, // 'X'
    0x82,0x44,0x44,0x28,0x28,0x10,0x10,0x10,0x10, // 'Y'
    0xFE,0x02,0x04,0x08,0x10,0x20,0x40,0x80,0xFE, // 'Z'
    0xD2,0x49,0x24,0x93,0x00, // '['
    0x84,0x10,0x84,0x10,0x84,0x10,0x80, // '\'
    0xC9,0x24,0x92,0x4B,0x00, // ']'
    0x18,0x12,0x10,0x80, // '^'
    0xFC, // '_'
    0x42, // '`'
    0x79,0x08,0x13,0xE8,0x51,0x9D,0x00, // 'a'
    0x81,0x02,0x07,0xCC,0xD0,0xA1,0x42,0xCD,0xF0, // 'b'
    0x73,0x28,0x20,0x83,0x27,0x00, // 'c'
    0x04,0x08,0x13,0xEC,0xD0,0xA1,0x42,0xCC,0xF8, // 'd'
    0x79,0x9A,0x17,0xE8,0x18,0x9E,0x00, // 'e'
    0x32,0x11,0xE4,0x21,0x08,0x42,0x00, // 'f'
    0x7D,0x9A,0x14,0x28,0x59,0x9F,0x02,0x4C,0x70, // 'g'
    0x81,0x02,0x05,0xCC,0x50,0xA1,0x42,0x85,0x08, // 'h'
    0x8A,0xAA,0x80, // 'i'
    0x41,0x24,0x92,0x49,0x60, // 'j'
    0x81,0x02,0x04,0x49,0x14,0x30,0x50,0x91,0x10, // 'k'
    0xAA,0xAA,0xA0, // 'l'
    0xF7,0x22,0x28,0x8A,0x22,0x88,0xA2,0x28,0x88, // 'm'
    0xB9,0x8A,0x14,0x28,0x50,0xA1,0x00, // 'n'
    0x79,0x9A,0x14,0x28,0x59,0x9E,0x00, // 'o'
    0xF9,0x9A,0x14,0x28,0x59,0xBE,0x40,0x81,0x00, // 'p'
    0x7D,0x9A,0x14,0x28,0x59,0x9F,0x02,0x04,0x08, // 'q'
    0xB6,0x21,0x08,0x42,0x00, // 'r'
    0x72,0x28,0x1C,0x0A,0x27,0x00, // 's'
    0x42,0x3C,0x84,0x21,0x08,0x70, // 't'
    0x85,0x0A,0x14,0x28,0x51,0x9D,0x00, // 'u'
    0x85,0x09,0x22,0x44,0x86,0x0C,0x00, // 'v'
    0x88,0xA2,0x25,0x51,0x54,0x55,0x08,0x82,0x20, // 'w'
    0x84,0x91,0x21,0x84,0x89,0x21,0x00, // 'x'
    0x85,0x09,0x22,0x42,0x86,0x04,0x08,0x21,0x80, // 'y'
    0xF8,0x21,0x08,0x42,0x0F,0x80, // 'z'
    0x38,0x82,0x08,0x23,0x02,0x08,0x20,0x83,0x80, // '{'
    0xAA,0xAA,0xAA, // '|'
    0xE0,0x82,0x08,0x20,0x62,0x08,0x20,0x8E,0x00 // '}'
};
const GFXglyph Dialog_plain_12Glyphs[] PROGMEM = {
// bitmapOffset, width, height, xAdvance, xOffset, yOffset
      {     0,   2,   1,   5,    0,   -1 }, // ' '
      {     1,   2,   9,   6,    2,   -9 }, // '!'
      {     4,   4,   3,   6,    1,   -9 }, // '"'
      {     6,   9,   8,  11,    1,   -8 }, // '#'
      {    15,   6,  11,   9,    2,   -9 }, // '$'
      {    24,  11,   9,  12,    0,   -9 }, // '%'
      {    37,   9,   9,  11,    1,   -9 }, // '&'
      {    48,   2,   3,   4,    1,   -9 }, // '''
      {    49,   4,  11,   6,    1,  -10 }, // '('
      {    55,   4,  11,   6,    1,  -10 }, // ')'
      {    61,   6,   6,   7,    1,   -9 }, // '*'
      {    66,   8,   7,  11,    1,   -7 }, // '+'
      {    73,   2,   3,   5,    1,   -2 }, // ','
      {    74,   4,   1,   5,    1,   -4 }, // '-'
      {    75,   2,   2,   5,    1,   -2 }, // '.'
      {    76,   5,  10,   5,    0,   -9 }, // '/'
      {    83,   7,   9,   9,    1,   -9 }, // '0'
      {    91,   6,   9,   9,    1,   -9 }, // '1'
      {    98,   7,   9,   9,    1,   -9 }, // '2'
      {   106,   7,   9,   9,    1,   -9 }, // '3'
      {   114,   7,   9,   9,    1,   -9 }, // '4'
      {   122,   7,   9,   9,    1,   -9 }, // '5'
      {   130,   7,   9,   9,    1,   -9 }, // '6'
      {   138,   7,   9,   9,    1,   -9 }, // '7'
      {   146,   7,   9,   9,    1,   -9 }, // '8'
      {   154,   7,   9,   9,    1,   -9 }, // '9'
      {   162,   2,   6,   5,    1,   -6 }, // ':'
      {   164,   2,   7,   5,    1,   -6 }, // ';'
      {   166,   9,   6,  11,    1,   -7 }, // '<'
      {   173,   9,   3,  11,    1,   -5 }, // '='
      {   177,   9,   6,  11,    1,   -7 }, // '>'
      {   184,   6,   9,   7,    0,   -9 }, // '?'
      {   191,  12,  11,  14,    1,   -9 }, // '@'
      {   208,   9,   9,   9,    0,   -9 }, // 'A'
      {   219,   7,   9,   9,    1,   -9 }, // 'B'
      {   227,   7,   9,   9,    1,   -9 }, // 'C'
      {   235,   8,   9,  10,    1,   -9 }, // 'D'
      {   244,   7,   9,   9,    1,   -9 }, // 'E'
      {   252,   6,   9,   8,    1,   -9 }, // 'F'
      {   259,   8,   9,  10,    1,   -9 }, // 'G'
      {   268,   8,   9,  10,    1,   -9 }, // 'H'
      {   277,   2,   9,   4,    1,   -9 }, // 'I'
      {   280,   4,  11,   4,   -1,   -9 }, // 'J'
      {   286,   8,   9,   8,    1,   -9 }, // 'K'
      {   295,   6,   9,   7,    1,   -9 }, // 'L'
      {   302,   9,   9,  11,    1,   -9 }, // 'M'
      {   313,   8,   9,  10,    1,   -9 }, // 'N'
      {   322,   8,   9,  10,    1,   -9 }, // 'O'
      {   331,   7,   9,   9,    1,   -9 }, // 'P'
      {   339,   8,  11,  10,    1,   -9 }, // 'Q'
      {   359,   7,   9,   9,    1,   -9 }, // ’S’
```

```
      {   350,   8,   9,   9,    1,   -9 }, // ’R’
      {   359,   7,   9,   9,    1,   -9 }, // ’S’
      {   367,   8,   9,   8,    0,   -9 }, // ’T’
      {   376,   8,   9,  10,    1,   -9 }, // ’U’
      {   385,  11,   9,   9,   -1,   -9 }, // ’V’
      {   398,  12,   9,  12,    0,   -9 }, // ’W’
      {   412,   8,   9,   8,    0,   -9 }, // ’X’
      {   421,   8,   9,   8,    0,   -9 }, // ’Y’
      {   430,   8,   9,  10,    1,   -9 }, // ’Z’
      {   439,   3,  11,   6,    2,   -9 }, // ’[’
      {   444,   5,  10,   5,    0,   -9 }, // ’\’
      {   451,   3,  11,   6,    1,   -9 }, // ’]’
      {   456,   9,   3,  11,    1,   -9 }, // ’^’
      {   460,   7,   1,   7,    0,    2 }, // ’_’
      {   461,   4,   2,   7,    1,  -10 }, // ’‘’
      {   462,   7,   7,   9,    1,   -7 }, // ’a’
      {   469,   7,  10,   9,    1,  -10 }, // ’b’
      {   478,   6,   7,   8,    1,   -7 }, // ’c’
      {   484,   7,  10,   9,    1,  -10 }, // ’d’
      {   493,   7,   7,   9,    1,   -7 }, // ’e’
      {   500,   5,  10,   5,    0,  -10 }, // ’f’
      {   507,   7,  10,   9,    1,   -7 }, // ’g’
      {   516,   7,  10,   9,    1,  -10 }, // ’h’
      {   525,   2,   9,   4,    1,   -9 }, // ’i’
      {   528,   3,  12,   4,    0,   -9 }, // ’j’
      {   533,   7,  10,   8,    1,  -10 }, // ’k’
      {   542,   2,  10,   4,    1,  -10 }, // ’l’
      {   545,  10,   7,  12,    1,   -7 }, // ’m’
      {   554,   7,   7,   9,    1,   -7 }, // ’n’
      {   561,   7,   7,   9,    1,   -7 }, // ’o’
      {   568,   7,  10,   9,    1,   -7 }, // ’p’
      {   577,   7,  10,   9,    1,   -7 }, // ’q’
      {   586,   5,   7,   6,    1,   -7 }, // ’r’
      {   591,   6,   7,   8,    1,   -7 }, // ’s’
      {   597,   5,   9,   6,    0,   -9 }, // ’t’
      {   603,   7,   7,   9,    1,   -7 }, // ’u’
      {   610,   7,   7,   7,    0,   -7 }, // ’v’
      {   617,  10,   7,  10,    0,   -7 }, // ’w’
      {   626,   7,   7,   7,    0,   -7 }, // ’x’
      {   633,   7,  10,   7,    0,   -7 }, // ’y’
      {   642,   6,   7,   6,    0,   -7 }, // ’z’
      {   648,   6,  11,   9,    2,   -9 }, // ’{’
      {   657,   2,  12,   5,    2,   -9 }, // ’|’
      {   660,   6,  11,   9,    1,   -9 } // ’}’
};
const GFXfont Dialog_plain_12 PROGMEM = {
(uint8_t  *)Dialog_plain_12Bitmaps,(GFXglyph *)Dialog_plain_12Glyphs,0x20, 0x7E, 15};
```

3.8 dialog9.h

```
// Created by https://oleddisplay.squix.ch/ Consider a donation
// In case of problems make sure that you are using the font file with the correct version!
const uint8_t Dialog_plain_9Bitmaps[] PROGMEM = {

    // Bitmap Data:
    0x00, // ’ ’
    0xAA,0x88, // ’!’
    0xAA, // ’"’
    0x28,0x53,0xF1,0x4F,0xCA,0x14,0x00, // ’#’
    0x21,0xEA,0x38,0x38,0xAF,0x08, // ’$’
    0xE4,0x52,0x2A,0x1F,0xE1,0x51,0x28,0x9C, // ’%’
    0x30,0x91,0x03,0x29,0x53,0x1B,0x00, // ’&’
    0xA0, // ”’
    0x52,0x49,0x22, // ’(’
    0x91,0x24,0xA4, // ’)’
    0xA9,0xC7,0x2A, // ’*’
    0x20,0x8F,0x88,0x20, // ’+’
    0xA0, // ’,’
    0xC0, // ’-’
    0x80, // ’.’
    0x22,0x44,0x48,0x80, // ’/’
    0x64,0xA5,0x29,0x49,0x80, // ’0’
    0xC4,0x44,0x44,0xE0, // ’1’
    0x64,0x84,0x44,0x43,0xC0, // ’2’
    0x64,0x84,0xC1,0x0B,0x80, // ’3’
    0x10,0xC5,0x14,0x93,0xE1,0x00, // ’4’
    0xF4,0x21,0xC1,0x0B,0x80, // ’5’
    0x76,0x21,0xC9,0x49,0x80, // ’6’
    0xF0,0x88,0x42,0x21,0x00, // ’7’
    0x64,0xA4,0xC9,0x49,0x80, // ’8’
    0x64,0xA5,0xE1,0x1B,0x80, // ’9’
    0x80,0x80, // ’:’
    0x80,0xA0, // ’;’
    0x04,0x73,0x01,0xC0,0x40, // '<'
    0xFC,0x03,0xF0, // '='
    0x80,0xE0,0x33,0x88,0x00, // '>'
    0xF0,0x88,0x84,0x01,0x00, // '?'
    0x3C,0x21,0x27,0x54,0xAA,0x54,0xF1,0x10,0x70, // '@'
    0x30,0x61,0x22,0x47,0x90,0xA1,0x00, // 'A'
    0xF2,0x28,0xBC,0x8A,0x2F,0x00, // 'B'
    0x73,0x28,0x20,0x83,0x07,0x80, // 'C'
    0xF2,0x68,0xA2,0x8A,0x6F,0x00, // 'D'
    0xF4,0x21,0xE8,0x43,0xC0, // 'E'
    0xF4,0x21,0xE8,0x42,0x00, // 'F'
    0x73,0x28,0x26,0x8B,0x27,0x00, // 'G'
    0x8A,0x28,0xBE,0x8A,0x28,0x80, // 'H'
    0xAA,0xA8, // 'I'
    0x49,0x24,0x92,0x80, // 'J'
    0x8A,0x4A,0x30,0xA2,0x48,0x80, // 'K'
    0x84,0x21,0x08,0x43,0xC0, // 'L'
    0x85,0x9B,0x35,0xAB,0x50,0xA1,0x00, // 'M'
    0x8B,0x2C,0xAA,0x9A,0x68,0x80, // 'N'
    0x73,0x68,0xA2,0x8B,0x67,0x00, // 'O'
    0xE4,0xA5,0xC8,0x42,0x00, // 'P'
    0x73,0x68,0xA2,0x8B,0x67,0x04, // 'Q'
    0xF2,0x49,0x38,0xA2,0x48,0x80, // 'R'
    0x72,0x28,0x1C,0x0A,0x27,0x00, // 'S'
    0xF8,0x82,0x08,0x20,0x82,0x00, // 'T'
    0x8A,0x28,0xA2,0x8A,0x27,0x00, // 'U'
    0x85,0x09,0x22,0x44,0x86,0x0C,0x00, // 'V'
    0x92,0x92,0x54,0x54,0x6C,0x28,0x28, // 'W'
    0xCC,0x90,0xC1,0x83,0x09,0x23,0x00, // 'X'
    0x89,0x45,0x08,0x20,0x82,0x00, // 'Y'
    0xF8,0x21,0x08,0x42,0x0F,0x80, // 'Z'
    0xD2,0x49,0x26, // '['
    0x88,0x44,0x42,0x20, // '\'
    0xC9,0x24,0x96, // ']'
    0x30,0x90, // '^'
    0xF8, // '_'
    0x88, // '`'
    0x70,0xBD,0x2F,0x00, // 'a'
    0x84,0x21,0xC9,0x4A,0x5C, // 'b'
    0x74,0x21,0x07,0x00, // 'c'
    0x10,0x84,0xE9,0x4A,0x4E, // 'd'
    0x64,0xBD,0x07,0x00, // 'e'
    0x72,0x11,0xC4,0x21,0x08, // 'f'
    0x74,0xA5,0x27,0x09,0x80, // 'g'
    0x84,0x21,0xE9,0x4A,0x52, // 'h'
    0x8A,0xA8, // 'i'
    0x41,0x24,0x92,0xC0, // 'j'
    0x84,0x21,0x2A,0x62,0x92, // 'k'
    0xAA,0xAA, // 'l'
    0xFE,0x92,0x92,0x92,0x92, // 'm'
    0xF4,0xA5,0x29,0x00, // 'n'
    0x64,0xA5,0x26,0x00, // 'o'
    0xE4,0xA5,0x2E,0x42,0x00, // 'p'
    0x74,0xA5,0x27,0x08,0x40, // 'q'
    0xE8,0x88,0x80, // 'r'
    0xE8,0x62,0xE0, // 's'
    0x47,0x90,0x84,0x38, // 't'
    0x94,0xA5,0x2F,0x00, // 'u'
    0x8A,0x25,0x14,0x20, // 'v'
    0x92,0xAA,0xAA,0x44,0x44, // 'w'
    0x89,0x42,0x14,0x88, // 'x'
    0x89,0x11,0x42,0x82,0x04,0x30,0x00, // 'y'
    0xF0,0x88,0x8F,0x00, // 'z'
    0x64,0x48,0x44,0x46, // '{'
    0xAA,0xAA,0x80, // '|'
    0xC4,0x42,0x44,0x4C // '}'
};
const GFXglyph Dialog_plain_9Glyphs[] PROGMEM = {
// bitmapOffset, width, height, xAdvance, xOffset, yOffset
      {     0,   2,   1,   4,    0,   -1 }, // ' '
      {     1,   2,   7,   4,    1,   -7 }, // '!'
      {     3,   4,   2,   5,    1,   -7 }, // '"'
      {     4,   7,   7,   9,    1,   -7 }, // '#'
      {    11,   6,   8,   7,    0,   -7 }, // '$'
      {    17,   9,   7,  10,    0,   -7 }, // '%'
      {    25,   7,   7,   9,    1,   -7 }, // '&'
      {    32,   2,   2,   3,    1,   -7 }, // '''
      {    33,   3,   8,   5,    1,   -8 }, // '('
      {    36,   3,   8,   5,    1,   -8 }, // ')'
      {    39,   6,   4,   6,    0,   -7 }, // '*'
      {    42,   6,   5,   9,    1,   -5 }, // '+'
      {    46,   2,   2,   4,    1,   -1 }, // ','
      {    47,   3,   1,   4,    1,   -3 }, // '-'
      {    48,   2,   1,   4,    1,   -1 }, // '.'
      {    49,   4,   7,   4,    0,   -7 }, // '/'
      {    53,   5,   7,   7,    1,   -7 }, // '0'
      {    58,   4,   7,   7,    2,   -7 }, // '1'
      {    62,   5,   7,   7,    1,   -7 }, // '2'
      {    67,   5,   7,   7,    1,   -7 }, // '3'
      {    72,   6,   7,   7,    1,   -7 }, // '4'
      {    78,   5,   7,   7,    1,   -7 }, // '5'
      {    83,   5,   7,   7,    1,   -7 }, // '6'
      {    88,   5,   7,   7,    1,   -7 }, // '7'
      {    93,   5,   7,   7,    1,   -7 }, // '8'
      {    98,   5,   7,   7,    1,   -7 }, // '9'
      {   103,   2,   5,   4,    1,   -5 }, // ':'
      {   105,   2,   6,   4,    1,   -5 }, // ';'
      {   107,   7,   5,   9,    1,   -5 }, // '<'
      {   112,   7,   3,   9,    1,   -4 }, // '='
      {   115,   7,   5,   9,    1,   -5 }, // '>'
      {   120,   5,   7,   6,    1,   -7 }, // '?'
      {   125,   9,   8,  11,    1,   -7 }, // '@'
      {   134,   7,   7,   7,    0,   -7 }, // 'A'
      {   141,   6,   7,   8,    1,   -7 }, // 'B'
      {   147,   6,   7,   8,    1,   -7 }, // 'C'
      {   153,   6,   7,   8,    1,   -7 }, // 'D'
      {   159,   5,   7,   7,    1,   -7 }, // 'E'
      {   164,   5,   7,   7,    1,   -7 }, // 'F'
      {   169,   6,   7,   8,    1,   -7 }, // 'G'
      {   175,   6,   7,   8,    1,   -7 }, // 'H'
      {   181,   2,   7,   4,    1,   -7 }, // 'I'
      {   183,   3,   9,   4,    0,   -7 }, // 'J'
      {   187,   6,   7,   7,    1,   -7 }, // 'K'
      {   193,   5,   7,   6,    1,   -7 }, // 'L'
      {   198,   7,   7,   9,    1,   -7 }, // 'M'
      {   205,   6,   7,   8,    1,   -7 }, // 'N'
      {   211,   6,   7,   8,    1,   -7 }, // 'O'
      {   217,   5,   7,   7,    1,   -7 }, // 'P'
      {   222,   6,   8,   8,    1,   -7 }, // 'Q'
      {   228,   6,   7,   7,    1,   -7 }, // 'R'
      {   234,   6,   7,   8,    1,   -7 }, // 'S'
      {   240,   6,   7,   6,    0,   -7 }, // 'T'
      {   246,   6,   7,   8,    1,   -7 }, // 'U'
      {   252,   7,   7,   7,    0,   -7 }, // 'V'
      {   259,   8,   7,   8,    0,   -7 }, // 'W'
      {   266,   7,   7,   7,    0,   -7 }, // 'X'
      {   273,   6,   7,   6,    0,   -7 }, // 'Y'
      {   279,   6,   7,   6,    0,   -7 }, // 'Z'
      {   285,   3,   8,   5,    1,   -7 }, // '['
      {   288,   4,   7,   4,    0,   -7 }, // '\'
      {   292,   3,   8,   5,    1,   -7 }, // ']'
      {   295,   7,   2,   9,    1,   -7 }, // '^'
      {   297,   6,   1,   6,    0,    1 }, // '_'
      {   298,   3,   2,   6,    1,   -8 }, // '`'
      {   299,   5,   5,   7,    1,   -5 }, // 'a'
      {   303,   5,   8,   7,    1,   -8 }, // 'b'
      {   308,   5,   5,   7,    1,   -5 }, // 'c'
      {   312,   5,   8,   7,    1,   -8 }, // 'd'
      {   317,   5,   5,   7,    1,   -5 }, // 'e'
      {   321,   5,   8,   4,    0,   -8 }, // 'f'
      {   326,   5,   7,   7,    1,   -5 }, // 'g'
      {   331,   5,   8,   7,    1,   -8 }, // 'h'
      {   336,   2,   7,   4,    1,   -7 }, // 'i'
      {   338,   3,   9,   4,    0,   -7 }, // 'j'
      {   342,   5,   8,   6,    1,   -8 }, // 'k'
      {   347,   2,   8,   4,    1,   -8 }, // 'l'
      {   349,   8,   5,  10,    1,   -5 }, // 'm'
      {   354,   5,   5,   7,    1,   -5 }, // 'n'
      {   358,   5,   5,   7,    1,   -5 }, // 'o'
      {   362,   5,   7,   7,    1,   -5 }, // 'p'
      {   367,   5,   7,   7,    1,   -5 }, // 'q'
      {   372,   4,   5,   5,    1,   -5 }, // 'r'
      {   375,   4,   5,   6,    1,   -5 }, // 's'
      {   378,   5,   6,   5,    0,   -6 }, // 't'
      {   382,   5,   5,   7,    1,   -5 }, // 'u'
      {   386,   6,   5,   6,    0,   -5 }, // 'v'
      {   390,   8,   5,   8,    0,   -5 }, // 'w'
      {   395,   6,   5,   6,    0,   -5 }, // 'x'
      {   399,   7,   7,   6,    0,   -5 }, // 'y'
      {   406,   5,   5,   7,    1,   -5 }, // 'z'
      {   410,   4,   8,   6,    1,   -7 }, // '{'
      {   414,   2,   9,   4,    1,   -7 }, // '|'
      {   417,   4,   8,   6,    1,   -7 } // '}'
};
const GFXfont Dialog_plain_9 PROGMEM = {
(uint8_t  *)Dialog_plain_9Bitmaps,(GFXglyph *)Dialog_plain_9Glyphs,0x20, 0x7E, 12};
```

3.9 gothic12.h

```
// Created by https://oleddisplay.squix.ch/ Consider a donation
// In case of problems make sure that you are using the font file with the correct version!
const uint8_t URW_Gothic_L_Book_12Bitmaps[] PROGMEM = {

    // Bitmap Data:
    0x00, // ’ ’
    0xAA,0xA8,0x80, // ’!’
    0xDB,0x00, // ’"’
    0x24,0x24,0xFE,0x48,0x48,0xFC,0x48,0x48,0x50, // ’#’
    0x21,0xC8,0xA0,0xC0,0xC0,0xA2,0x89,0xC2,0x00, // ’$’
    0x62,0x25,0x09,0x41,0xA0,0x08,0x04,0xC1,0x48,0x92,0x23,0x00, // ’%’
    0x38,0x24,0x12,0x07,0x27,0x24,0x72,0x11,0x96,0x79,0x00, // ’&’
    0xA8, // ’’’
    0x32,0x11,0x08,0x42,0x10,0x41,0x04, // ’(’
    0xC1,0x08,0x21,0x08,0x46,0x22,0x20, // ’)’
    0x04,0xEC, // ’*’
    0x10,0x10,0xFE,0x10,0x10,0x10, // ’+’
    0x4A,0x00, // ’,’
    0xF0, // ’-’
    0x80, // ’.’
    0x08,0x41,0x04,0x20,0x82,0x10,0x41,0x08,0x00, // ’/’
    0x79,0x9A,0x14,0x28,0x50,0xA1,0x66,0x78, // ’0’
    0xE2,0x22,0x22,0x22,0x20, // ’1’
    0x79,0x9A,0x10,0x20,0x82,0x08,0x20,0xFC, // ’2’
    0x70,0x90,0x21,0x80,0xC0,0xA1,0x66,0x78, // ’3’
    0x0C,0x0C,0x14,0x24,0x24,0x44,0xFE,0x04,0x04, // ’4’
    0x78,0x81,0xE6,0x40,0x40,0xA1,0x64,0x78, // ’5’
    0x10,0x40,0xE2,0x68,0x50,0xA1,0x66,0x78, // ’6’
    0xF8,0x21,0x04,0x20,0x82,0x10,0x40, // ’7’
    0x71,0x12,0x23,0x8C,0xD0,0xA1,0x66,0x78, // ’8’
    0x79,0x9A,0x14,0x2C,0xCF,0x04,0x08,0x20, // ’9’
    0x80,0x20, // ’:’
    0x40,0x05,0x20, // ’;’
    0x04,0x31,0x86,0x03,0x01,0x80,0x00, // ’<’
    0xFC,0x03,0xF0, // ’=’
    0x80,0xC0,0x60,0x63,0x18,0x00,0x00, // ’>’
    0x72,0x68,0x82,0x10,0xC2,0x00,0x20, // ’?’
    0x3E,0x1B,0xCD,0x2A,0x92,0xA4,0xA9,0x4F,0xF1,0x84,0x3E,0x00, // ’@’
    0x08,0x05,0x01,0x40,0x90,0x22,0x0F,0x84,0x11,0x04,0x80,0x80, // ’A’
    0xF1,0x12,0x24,0x4F,0x11,0xA1,0x42,0xF8, // ’B’
    0x3E,0x18,0x4C,0x0A,0x00,0x80,0x20,0x0C,0x09,0x84,0x3E,0x00, // ’C’
    0xF8,0x43,0x20,0xD0,0x28,0x14,0x0A,0x0D,0x0C,0xFC,0x00, // ’D’
    0xFA,0x08,0x20,0xFA,0x08,0x20,0xF8, // ’E’
    0xFA,0x08,0x20,0xF2,0x08,0x20,0x80, // ’F’
    0x3E,0x0C,0x33,0x00,0x40,0x08,0xF9,0x01,0x10,0x23,0x08,0x1E,0x00, // ’G’
    0x85,0x0A,0x14,0x2F,0xD0,0xA1,0x42,0x84, // ’H’
    0xAA,0xAA,0x80, // ’I’
    0x08,0x20,0x82,0x08,0x20,0xA2,0x70, // ’J’
    0x85,0x12,0x45,0x0A,0x1E,0x24,0x44,0x84, // ’K’
    0x82,0x08,0x20,0x82,0x08,0x20,0xF8, // ’L’
    0xC1,0xB0,0x6C,0x2A,0x8A,0xA2,0xA5,0x29,0x4A,0x62,0x88,0x80, // ’M’
    0xC2,0xC2,0xA2,0xB2,0x92,0x8A,0x8A,0x86,0x82, // ’N’
    0x3E,0x18,0xCC,0x1A,0x02,0x80,0xA0,0x2C,0x19,0x8C,0x3E,0x00, // ’O’
    0xF9,0x0A,0x14,0x2F,0x90,0x20,0x40,0x80, // ’P’
    0x1E,0x0C,0x33,0x03,0x40,0x28,0x05,0xF8,0x91,0xA3,0x0C,0x1E,0xC0, // ’Q’
    0xF9,0x1A,0x14,0x28,0xD6,0x24,0x44,0x88, // ’R’
    0x72,0x28,0x20,0x70,0x28,0xA2,0x70, // ’S’
    0xF8,0x82,0x08,0x20,0x82,0x08,0x20, // ’T’
    0x85,0x0A,0x14,0x28,0x50,0xA1,0x66,0x78, // ’U’
    0x81,0x20,0x90,0x88,0x42,0x41,0x20,0x50,0x30,0x08,0x00, // ’V’
    0x84,0x28,0x62,0x46,0x24,0xA4,0x4A,0x42,0x94,0x31,0x43,0x18,0x10,0x80, // ’W’
    0x42,0x44,0x28,0x18,0x10,0x28,0x24,0x44,0x82, // ’X’
    0x82,0x44,0x44,0x28,0x28,0x10,0x10,0x10,0x10, // ’Y’
    0xFC,0x10,0x60,0x82,0x04,0x10,0x40,0xFC, // ’Z’
    0x68,0x88,0x88,0x88,0x84,0x20, // ’[’
    0x82,0x04,0x10,0x20,0x81,0x04,0x10, // ’\’
    0xC2,0x22,0x22,0x22,0x24,0x80, // ’]’
    0x10,0x28,0x28,0x24,0x44,0x44,0x82, // ’^’
    0xFC, // ’_’
    0x84, // ’‘’
    0x3A,0x46,0x82,0x82,0x82,0x46,0x3A, // ’a’
    0x80,0x80,0xB8,0xC4,0x82,0x82,0x82,0xC4,0xB8, // ’b’
    0x38,0x44,0x80,0x80,0x80,0x44,0x38, // ’c’
    0x02,0x02,0x3A,0x46,0x82,0x82,0x82,0x46,0x3A, // ’d’
    0x38,0x44,0x82,0xFE,0x80,0x44,0x38, // ’e’
    0x32,0x3C,0x84,0x21,0x08,0x40, // ’f’
    0x3A,0x46,0x82,0x82,0x82,0x46,0x3A,0xC6,0x3C, // ’g’
    0x81,0x02,0xE6,0x68,0x50,0xA1,0x42,0x84, // ’h’
    0x8A,0xAA,0x80, // ’i’
    0x41,0x24,0x92,0x4A,0x00, // ’j’
    0x82,0x08,0xA4,0xA2,0x8E,0x24,0x88, // ’k’
    0xAA,0xAA,0x80, // ’l’
    0xF7,0x22,0x28,0x8A,0x22,0x88,0xA2,0x28,0x88, // ’m’
    0xB9,0x9A,0x14,0x28,0x50,0xA1,0x00, // ’n’
    0x38,0x44,0x82,0x82,0x82,0x44,0x38, // ’o’
    0xB8,0xC4,0x82,0x82,0x82,0xC4,0xB8,0x80,0x80, // ’p’
    0x3A,0x46,0x82,0x82,0x82,0x46,0x3A,0x02,0x02, // ’q’
    0xE8,0x88,0x88,0x80, // ’r’
    0x64,0xA0,0xC1,0x4B,0x80, // 's'
    0x44,0xE4,0x44,0x44,0x40, // 't'
    0x85,0x0A,0x14,0x28,0x59,0x9D,0x00, // 'u'
    0x84,0x44,0x44,0x48,0x28,0x30,0x10, // 'v'
    0x80,0x51,0x91,0x32,0x26,0x43,0x50,0x66,0x0C,0xC0, // 'w'
    0x88,0x90,0xC1,0x03,0x09,0x22,0x00, // 'x'
    0x84,0x89,0x22,0x43,0x06,0x04,0x10,0x20, // 'y'
    0xF8,0x42,0x08,0x42,0x0F,0x80, // 'z'
    0x64,0x44,0x48,0x44,0x44,0x60, // '{'
    0xAA,0xAA,0x80, // '|'
    0xC4,0x44,0x42,0x44,0x44,0xC0 // '}'
};
const GFXglyph URW_Gothic_L_Book_12Glyphs[] PROGMEM = {
// bitmapOffset, width, height, xAdvance, xOffset, yOffset
      {     0,    2,    1,    4,     0,    -1 }, // ' '
      {     1,    2,    9,    4,     1,    -9 }, // '!'
      {     4,    3,    3,    5,     1,    -9 }, // '"'
      {     6,    8,    9,    8,     0,    -9 }, // '#'
      {    15,    6,   11,    8,     1,   -10 }, // '$'
      {    24,   10,    9,   12,     1,    -9 }, // '%'
      {    36,    9,    9,   11,     1,    -9 }, // '&'
      {    47,    2,    3,    4,     1,    -9 }, // '''
      {    48,    5,   11,    6,     1,    -9 }, // '('
      {    55,    5,   11,    6,     0,    -9 }, // ')'
      {    62,    4,    4,    6,     1,   -10 }, // '*'
      {    64,    8,    6,    8,     0,    -7 }, // '+'
      {    70,    3,    3,    4,     1,    -2 }, // ','
      {    72,    5,    1,    5,     0,    -4 }, // '-'
      {    73,    2,    1,    4,     1,    -1 }, // '.'
      {    74,    6,   11,    6,     0,    -9 }, // '/'
      {    83,    7,    9,    8,     1,    -9 }, // '0'
      {    91,    4,    9,    8,     1,    -9 }, // '1'
      {    96,    7,    9,    8,     1,    -9 }, // '2'
      {   104,    7,    9,    8,     1,    -9 }, // '3'
      {   112,    8,    9,    8,     0,    -9 }, // '4'
      {   121,    7,    9,    8,     0,    -9 }, // '5'
      {   129,    7,    9,    8,     1,    -9 }, // '6'
      {   137,    6,    9,    8,     1,    -9 }, // '7'
      {   144,    7,    9,    8,     1,    -9 }, // '8'
      {   152,    7,    9,    8,     1,    -9 }, // '9'
      {   160,    2,    6,    4,     1,    -6 }, // ':'
      {   162,    3,    7,    4,     0,    -6 }, // ';'
      {   165,    7,    7,    9,     1,    -7 }, // '<'
      {   172,    7,    3,    8,     1,    -5 }, // '='
      {   175,    7,    7,    9,     1,    -7 }, // '>'
      {   182,    6,    9,    8,     1,    -9 }, // '?'
      {   189,   10,    9,   12,     1,    -9 }, // '@'
      {   201,   10,    9,   10,     0,    -9 }, // 'A'
      {   213,    7,    9,    9,     1,    -9 }, // 'B'
      {   221,   10,    9,   11,     1,    -9 }, // 'C'
      {   233,    9,    9,   11,     1,    -9 }, // 'D'
      {   244,    6,    9,    8,     1,    -9 }, // 'E'
      {   251,    6,    9,    7,     1,    -9 }, // 'F'
      {   258,   11,    9,   12,     1,    -9 }, // 'G'
      {   271,    7,    9,    9,     1,    -9 }, // 'H'
      {   279,    2,    9,    4,     1,    -9 }, // 'I'
      {   282,    6,    9,    7,     0,    -9 }, // 'J'
      {   289,    7,    9,    8,     1,    -9 }, // 'K'
      {   297,    6,    9,    7,     1,    -9 }, // 'L'
      {   304,   10,    9,   12,     1,    -9 }, // 'M'
      {   316,    8,    9,   10,     1,    -9 }, // 'N'
      {   325,   10,    9,   12,     1,    -9 }, // 'O'
      {   337,    7,    9,    9,     1,    -9 }, // 'P'
      {   345,   11,    9,   13,     1,    -9 }, // 'Q'
      {   358,    7,    9,    9,     1,    -9 }, // 'R'
      {   366,    6,    9,    8,     1,    -9 }, // 'S'
      {   373,    6,    9,    6,     0,    -9 }, // 'T'
      {   380,    7,    9,    9,     1,    -9 }, // 'U'
      {   388,    9,    9,    9,     0,    -9 }, // 'V'
      {   399,   12,    9,   13,     0,    -9 }, // 'W'
      {   413,    8,    9,    8,     0,    -9 }, // 'X'
      {   422,    8,    9,    8,     0,    -9 }, // 'Y'
      {   431,    7,    9,    8,     1,    -9 }, // 'Z'
      {   439,    4,   11,    5,     1,    -9 }, // '['
      {   445,    6,    9,    8,     1,    -9 }, // '\'
      {   452,    4,   11,    5,     0,    -9 }, // ']'
      {   458,    8,    7,    8,     0,    -9 }, // '^'
      {   465,    7,    1,    7,     0,     1 }, // '_'
      {   466,    4,    2,    6,     1,   -10 }, // '`'
      {   467,    8,    7,   10,     1,    -7 }, // 'a'
      {   474,    8,    9,   10,     1,    -9 }, // 'b'
      {   483,    8,    7,    9,     1,    -7 }, // 'c'
      {   490,    8,    9,   10,     1,    -9 }, // 'd'
      {   499,    8,    7,   10,     1,    -7 }, // 'e'
      {   506,    5,    9,    5,     0,    -9 }, // 'f'
      {   512,    8,    9,   10,     1,    -7 }, // 'g'
      {   521,    7,    9,    9,     1,    -9 }, // 'h'
```

```
     {   529,   2,   9,   4,     1,   -9 }, // 'i'
     {   532,   3,  11,   4,     0,   -9 }, // 'j'
     {   537,   6,   9,   7,     1,   -9 }, // 'k'
     {   544,   2,   9,   4,     1,   -9 }, // 'l'
     {   547,  10,   7,  12,     1,   -7 }, // 'm'
     {   556,   7,   7,   9,     1,   -7 }, // 'n'
     {   563,   8,   7,  10,     1,   -7 }, // 'o'
     {   570,   8,   9,  10,     1,   -7 }, // 'p'
     {   579,   8,   9,  10,     1,   -7 }, // 'q'
     {   588,   4,   7,   5,     1,   -7 }, // 'r'
     {   592,   5,   7,   7,     1,   -7 }, // 's'
     {   597,   4,   9,   6,     1,   -9 }, // 't'
     {   602,   7,   7,   9,     1,   -7 }, // 'u'
     {   609,   8,   7,   8,     0,   -7 }, // 'v'
     {   616,  11,   7,  11,     0,   -7 }, // 'w'
     {   626,   7,   7,   7,     0,   -7 }, // 'x'
     {   633,   7,   9,   7,     0,   -7 }, // 'y'
     {   641,   6,   7,   7,     1,   -7 }, // 'z'
     {   647,   4,  11,   5,     0,   -9 }, // '{'
     {   653,   2,   9,  10,     4,   -9 }, // '|'
     {   656,   4,  11,   5,     0,   -9 } // '}'
};
const GFXfont URW_Gothic_L_Book_12 PROGMEM = {
(uint8_t  *)URW_Gothic_L_Book_12Bitmaps,(GFXglyph *)URW_Gothic_L_Book_12Glyphs,0x20, 0x7E, 17};
```

3.10 gothic9.h

```
// Created by https://oleddisplay.squix.ch/ Consider a donation
// In case of problems make sure that you are using the font file with the correct version!
const uint8_t URW_Gothic_L_Book_9Bitmaps[] PROGMEM = {

    // Bitmap Data:
    0x00, // ' '
    0xAA,0x88, // '!'
    0xD8, // '"'
    0x51,0x4F,0x94,0xFA,0x8A,0x00, // '#'
    0x45,0xA0,0xC1,0x49,0x80, // '$'
    0xC8,0xB0,0xB0,0x60,0x2C,0x32,0x1C, // '%'
    0x70,0xA1,0xD3,0x49,0x91,0x1D,0x00, // '&'
    0xA0, // '''
    0x04,0x88,0x88,0x42, // '('
    0x04,0x22,0x22,0x48, // ')'
    0xDB,0x00, // '*'
    0x20,0x8F,0x88,0x20, // '+'
    0x20, // ','
    0xE0, // '-'
    0x80, // '.'
    0x21,0x08,0x84,0x22,0x10, // '/'
    0x72,0x28,0xA2,0x8A,0x27,0x00, // '0'
    0xC9,0x24,0x90, // '1'
    0x72,0x20,0x84,0x21,0x0F,0x80, // '2'
    0x64,0x84,0x41,0x49,0x80, // '3'
    0x08,0x30,0xA2,0x44,0x9F,0x82,0x00, // '4'
    0x72,0x07,0x02,0x0A,0x27,0x00, // '5'
    0x21,0x07,0x22,0x8A,0x27,0x00, // '6'
    0xF0,0x88,0x44,0x21,0x00, // '7'
    0x64,0xA4,0xC9,0x49,0x80, // '8'
    0x72,0x28,0xA2,0x70,0x82,0x00, // '9'
    0x80,0x80, // ':'
    0x40,0x28, // ';'
    0x19,0x88,0x1C,0x08, // '<'
    0xF8,0x0F,0x80, // '='
    0x81,0x81,0x98,0x80, // '>'
    0x72,0x20,0x84,0x20,0x02,0x00, // '?'
    0x3C,0x52,0xAA,0xCA,0xD2,0x6C,0x38, // '@'
    0x10,0x30,0x28,0x48,0x7C,0x84,0x84, // 'A'
    0xE4,0xA5,0xC9,0x4B,0xC0, // 'B'
    0x3C,0x42,0x80,0x80,0x80,0x42,0x3C, // 'C'
    0xF1,0x12,0x14,0x28,0x51,0x3C,0x00, // 'D'
    0xF4,0x21,0xE8,0x43,0xC0, // 'E'
    0xF4,0x21,0xE8,0x42,0x00, // 'F'
    0x38,0x44,0x80,0x80,0x9E,0xC2,0x3C, // 'G'
    0x8A,0x28,0xBE,0x8A,0x28,0x80, // 'H'
    0xAA,0xA8, // 'I'
    0x10,0x84,0x21,0x49,0x80, // 'J'
    0x8A,0x4A,0x30,0xA2,0x48,0x80, // 'K'
    0x84,0x21,0x08,0x43,0xC0, // 'L'
    0xC6,0xC6,0xCA,0xAA,0xAA,0xB2,0x92, // 'M'
    0x8B,0x2C,0xAA,0xAA,0x68,0x80, // 'N'
    0x38,0x44,0x82,0x82,0x82,0x44,0x38, // 'O'
    0xF2,0x28,0xA2,0xF2,0x08,0x00, // 'P'
    0x38,0x44,0x82,0x82,0xF2,0x8C,0x7E, // 'Q'
    0xF2,0x28,0xA6,0xA2,0x49,0x80, // 'R'
    0x64,0xA0,0xC1,0x4B,0x80, // 'S'
    0xF8,0x82,0x08,0x20,0x82,0x00, // 'T'
    0x8A,0x28,0xA2,0x8A,0x27,0x00, // 'U'
    0x85,0x09,0x22,0x43,0x06,0x04,0x00, // 'V'
    0x89,0x22,0x49,0x91,0x54,0x56,0x19,0x82,0x20, // 'W'
    0x89,0x43,0x08,0x51,0x48,0x80, // 'X'
    0x8A,0x25,0x14,0x20,0x82,0x00, // 'Y'
    0xF0,0x88,0x84,0x43,0xC0, // 'Z'
    0x1A,0x49,0x22, // '['
    0x88,0x84,0x42,0x20, // '\'
    0x19,0x24,0x94, // ']'
    0x21,0x45,0x12,0x88, // '^'
    0xF8, // '_'
    0x80, // '`'
    0x7A,0x28,0xA2,0x78, // 'a'
    0x82,0x0F,0x22,0x8A,0x2F,0x00, // 'b'
    0x7A,0x08,0x20,0x78, // 'c'
    0x08,0x27,0xA2,0x8A,0x27,0x80, // 'd'
    0x72,0x2F,0xA0,0x78, // 'e'
    0x24,0xE4,0x44,0x40, // 'f'
    0x7A,0x28,0xA2,0x7A,0x27,0x00, // 'g'
    0x82,0x0F,0x22,0x8A,0x28,0x80, // 'h'
    0x8A,0xA8, // 'i'
    0x41,0x24,0x92,0x80, // 'j'
    0x84,0x25,0x4C,0x52,0x40, // 'k'
    0xAA,0xA8, // 'l'
    0xEC,0x92,0x92,0x92,0x92, // 'm'
    0xF2,0x28,0xA2,0x88, // 'n'
    0x72,0x28,0xA2,0x70, // 'o'
    0xF2,0x28,0xA2,0xF2,0x08,0x00, // 'p'
    0x7A,0x28,0xA2,0x78,0x20,0x80, // 'q'
    0xD2,0x48, // 'r'
    0xE8,0x4A,0xE0, // 's'
    0x44,0xE4,0x44,0x40, // 't'
    0x8A,0x28,0xA2,0x78, // 'u'
    0x8A,0x45,0x18,0x20, // 'v'
    0x82,0x92,0x6C,0x6C,0x24, // 'w'
    0x93,0x08,0xC9,0x00, // 'x'
    0x8A,0x45,0x18,0x20,0x84,0x00, // 'y'
    0xF1,0x11,0x0F,0x00, // 'z'
    0x64,0x4C,0x44,0x42, // '{'
    0xAA,0xA8, // '|'
    0xC4,0x46,0x44,0x48 // '}'
};
const GFXglyph URW_Gothic_L_Book_9Glyphs[] PROGMEM = {
// bitmapOffset, width, height, xAdvance, xOffset, yOffset
      {     0,   2,   1,   4,    0,   -1 }, // ' '
      {     1,   2,   7,   4,    1,   -7 }, // '!'
      {     3,   3,   2,   5,    1,   -7 }, // '"'
      {     4,   6,   7,   6,    0,   -7 }, // '#'
      {    10,   5,   7,   7,    1,   -7 }, // '$'
      {    15,   8,   7,  10,    1,   -7 }, // '%'
      {    22,   7,   7,   8,    1,   -7 }, // '&'
      {    29,   2,   2,   4,    1,   -7 }, // '''
      {    30,   4,   8,   5,    1,   -7 }, // '('
      {    34,   4,   8,   5,    0,   -7 }, // ')'
      {    38,   3,   3,   5,    1,   -7 }, // '*'
      {    40,   6,   5,   6,    0,   -5 }, // '+'
      {    44,   2,   2,   4,    1,   -1 }, // ','
      {    45,   4,   1,   4,    0,   -3 }, // '-'
      {    46,   2,   1,   4,    1,   -1 }, // '.'
      {    47,   5,   8,   5,    0,   -7 }, // '/'
      {    52,   6,   7,   6,    1,   -7 }, // '0'
      {    58,   3,   7,   6,    2,   -7 }, // '1'
      {    61,   6,   7,   6,    0,   -7 }, // '2'
      {    67,   5,   7,   6,    0,   -7 }, // '3'
      {    72,   7,   7,   6,    0,   -7 }, // '4'
      {    79,   6,   7,   6,    0,   -7 }, // '5'
      {    85,   6,   7,   6,    1,   -7 }, // '6'
      {    91,   5,   7,   6,    0,   -7 }, // '7'
      {    96,   5,   7,   6,    1,   -7 }, // '8'
      {   101,   6,   7,   6,    1,   -7 }, // '9'
      {   107,   2,   5,   4,    1,   -5 }, // ':'
      {   109,   3,   5,   4,    0,   -5 }, // ';'
      {   111,   6,   5,   7,    1,   -5 }, // '<'
      {   115,   6,   3,   6,    0,   -4 }, // '='
      {   118,   6,   5,   7,    1,   -5 }, // '>'
      {   122,   6,   7,   7,    0,   -7 }, // '?'
      {   128,   8,   7,  10,    1,   -7 }, // '@'
      {   135,   8,   7,   8,    0,   -7 }, // 'A'
      {   142,   5,   7,   7,    1,   -7 }, // 'B'
      {   147,   8,   7,   9,    1,   -7 }, // 'C'
      {   154,   7,   7,   9,    1,   -7 }, // 'D'
      {   161,   5,   7,   6,    1,   -7 }, // 'E'
      {   166,   5,   7,   6,    1,   -7 }, // 'F'
      {   171,   8,   7,  10,    1,   -7 }, // 'G'
      {   178,   6,   7,   8,    1,   -7 }, // 'H'
```

```
	  {   184,   2,   7,   4,    1,   -7 }, // 'I'
	  {   186,   5,   7,   6,    0,   -7 }, // 'J'
	  {   191,   6,   7,   7,    1,   -7 }, // 'K'
	  {   197,   5,   7,   6,    1,   -7 }, // 'L'
	  {   202,   8,   7,  10,    1,   -7 }, // 'M'
	  {   209,   6,   7,   8,    1,   -7 }, // 'N'
	  {   215,   8,   7,  10,    1,   -7 }, // 'O'
	  {   222,   6,   7,   8,    1,   -7 }, // 'P'
	  {   228,   8,   7,  10,    1,   -7 }, // 'Q'
	  {   235,   6,   7,   8,    1,   -7 }, // 'R'
	  {   241,   5,   7,   7,    1,   -7 }, // 'S'
	  {   246,   6,   7,   6,    0,   -7 }, // 'T'
	  {   252,   6,   7,   8,    1,   -7 }, // 'U'
	  {   258,   7,   7,   7,    0,   -7 }, // 'V'
	  {   265,  10,   7,  10,    0,   -7 }, // 'W'
	  {   274,   6,   7,   6,    0,   -7 }, // 'X'
	  {   280,   6,   7,   6,    0,   -7 }, // 'Y'
	  {   286,   5,   7,   7,    1,   -7 }, // 'Z'
	  {   291,   3,   8,   5,    1,   -7 }, // '['
	  {   294,   4,   7,   6,    1,   -7 }, // '\'
	  {   298,   3,   8,   5,    1,   -7 }, // ']'
	  {   301,   6,   5,   6,    0,   -7 }, // '^'
	  {   305,   6,   1,   6,    0,    0 }, // '_'
	  {   306,   3,   2,   4,    1,   -7 }, // '`'
	  {   307,   6,   5,   8,    1,   -5 }, // 'a'
	  {   311,   6,   7,   8,    1,   -7 }, // 'b'
	  {   317,   6,   5,   8,    1,   -5 }, // 'c'
	  {   321,   6,   7,   8,    1,   -7 }, // 'd'
	  {   327,   6,   5,   8,    1,   -5 }, // 'e'
	  {   331,   4,   7,   4,    0,   -7 }, // 'f'
	  {   335,   6,   7,   8,    1,   -5 }, // 'g'
	  {   341,   6,   7,   8,    1,   -7 }, // 'h'
	  {   347,   2,   7,   4,    1,   -7 }, // 'i'
	  {   349,   3,   9,   4,    0,   -7 }, // 'j'
	  {   353,   5,   7,   6,    1,   -7 }, // 'k'
	  {   358,   2,   7,   4,    1,   -7 }, // 'l'
	  {   360,   8,   5,  10,    1,   -5 }, // 'm'
	  {   365,   6,   5,   8,    1,   -5 }, // 'n'
	  {   369,   6,   5,   8,    1,   -5 }, // 'o'
	  {   373,   6,   7,   8,    1,   -5 }, // 'p'
	  {   379,   6,   7,   8,    1,   -5 }, // 'q'
	  {   385,   3,   5,   4,    1,   -5 }, // 'r'
	  {   387,   4,   5,   6,    1,   -5 }, // 's'
	  {   390,   4,   7,   4,    0,   -7 }, // 't'
	  {   394,   6,   5,   8,    1,   -5 }, // 'u'
	  {   398,   6,   5,   6,    0,   -5 }, // 'v'
	  {   402,   8,   5,   8,    0,   -5 }, // 'w'
	  {   407,   5,   5,   5,    0,   -5 }, // 'x'
	  {   411,   6,   7,   6,    0,   -5 }, // 'y'
	  {   417,   5,   5,   6,    1,   -5 }, // 'z'
	  {   421,   4,   8,   4,    0,   -7 }, // '{'
	  {   425,   2,   7,   8,    3,   -7 }, // '|'
	  {   427,   4,   8,   4,    0,   -7 } // '}'
};
const GFXfont URW_Gothic_L_Book_9 PROGMEM = {
(uint8_t  *)URW_Gothic_L_Book_9Bitmaps,(GFXglyph *)URW_Gothic_L_Book_9Glyphs,0x20, 0x7E, 13};
```

3.11 mono12.h

```
// Created by https://oleddisplay.squix.ch/ Consider a donation
// In case of problems make sure that you are using the font file with the correct version!
const uint8_t Monospaced_plain_12Bitmaps[] PROGMEM = {

	// Bitmap Data:
	0x00, // ' '
	0xAA,0xA2,0x80, // '!'
	0xAA,0xA0, // '"'
	0x14,0x24,0x7E,0x28,0x28,0xFC,0x48,0x50, // '#'
	0x21,0xCA,0xA8,0xE0,0xE2,0xAA,0x70,0x82,0x00, // '$'
	0x60,0x90,0x90,0x64,0x18,0x6C,0x12,0x12,0x0C, // '%'
	0x38,0x81,0x03,0x06,0x12,0xA7,0x64,0x74, // '&'
	0xA8, // '''
	0x64,0x48,0x88,0x88,0x44,0x60, // '('
	0xC4,0x42,0x22,0x22,0x44,0xC0, // ')'
	0x22,0xA7,0x1C,0xA8,0x80, // '*'
	0x10,0x10,0x10,0xFE,0x10,0x10,0x10, // '+'
	0x4A,0x00, // ','
	0xE0, // '-'
	0xA0, // '.'
	0x04,0x10,0x20,0x81,0x04,0x08,0x20,0x41,0x00, // '/'
	0x78,0x92,0x14,0x29,0x50,0xA1,0x24,0x78, // '0'
	0xE0,0x82,0x08,0x20,0x82,0x08,0xF8, // '1'
	0x79,0x08,0x10,0x20,0x82,0x08,0x20,0xFC, // '2'
```

```
    0x79,0x08,0x10,0x23,0x80,0x81,0x42,0x78, // '3'
    0x18,0x30,0xA3,0x44,0x91,0x3F,0x04,0x08, // '4'
    0xF9,0x02,0x07,0xC0,0xC0,0x81,0x46,0x78, // '5'
    0x38,0x8A,0x05,0xCC,0xD0,0xA1,0x26,0x78, // '6'
    0xFC,0x18,0x20,0x41,0x02,0x08,0x10,0x40, // '7'
    0x79,0x0A,0x14,0x27,0x90,0xA1,0x42,0x78, // '8'
    0x79,0x92,0x14,0x28,0xCE,0x81,0x44,0x70, // '9'
    0xA0,0xA0, // ':'
    0x48,0x04,0xA0, // ';'
    0x04,0x73,0x06,0x03,0x80,0x80, // '<'
    0xFC,0x03,0xF0, // '='
    0x80,0xE0,0x30,0x67,0x10,0x00, // '>'
    0x72,0x20,0x8C,0x61,0x00,0x10,0x40, // '?'
    0x38,0x9A,0x14,0xEA,0x54,0xA7,0x60,0x40,0x70, // '@'
    0x30,0x60,0xC2,0x44,0x89,0x1E,0x42,0x84, // 'A'
    0xF9,0x0A,0x14,0x2F,0x90,0xA1,0x42,0xF8, // 'B'
    0x38,0x8A,0x04,0x08,0x10,0x20,0x22,0x38, // 'C'
    0xF1,0x12,0x14,0x28,0x50,0xA1,0x44,0xF0, // 'D'
    0xFD,0x02,0x04,0x0F,0xD0,0x20,0x40,0xFC, // 'E'
    0xFD,0x02,0x04,0x0F,0xD0,0x20,0x40,0x80, // 'F'
    0x38,0x8A,0x04,0x08,0xD0,0xA1,0x22,0x38, // 'G'
    0x85,0x0A,0x14,0x2F,0xD0,0xA1,0x42,0x84, // 'H'
    0xF8,0x82,0x08,0x20,0x82,0x08,0xF8, // 'I'
    0x38,0x20,0x82,0x08,0x20,0xA2,0x70, // 'J'
    0x85,0x12,0x45,0x0E,0x12,0x26,0x44,0x84, // 'K'
    0x81,0x02,0x04,0x08,0x10,0x20,0x40,0xFC, // 'L'
    0x85,0x9B,0x35,0xAB,0x56,0xA1,0x42,0x84, // 'M'
    0xC5,0x8A,0x95,0x2B,0x52,0xA5,0x46,0x8C, // 'N'
    0x78,0x92,0x14,0x28,0x50,0xA1,0x24,0x78, // 'O'
    0xF9,0x0A,0x14,0x2F,0x90,0x20,0x40,0x80, // 'P'
    0x78,0x92,0x14,0x28,0x50,0xA1,0x26,0x78,0x10,0x20, // 'Q'
    0xF8,0x84,0x84,0x84,0xF8,0x88,0x84,0x84,0x82, // 'R'
    0x79,0x0A,0x06,0x07,0x80,0x81,0x42,0x78, // 'S'
    0xFE,0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x10, // 'T'
    0x85,0x0A,0x14,0x28,0x50,0xA1,0x42,0x78, // 'U'
    0x85,0x09,0x22,0x44,0x89,0x0C,0x18,0x30, // 'V'
    0x82,0x92,0x92,0xAA,0xAA,0xAA,0x6C,0x44,0x44, // 'W'
    0x84,0x91,0x21,0x83,0x06,0x12,0x24,0x84, // 'X'
    0x82,0x44,0x28,0x28,0x10,0x10,0x10,0x10,0x10, // 'Y'
    0xFC,0x18,0x20,0x83,0x04,0x10,0x60,0xFC, // 'Z'
    0xD2,0x49,0x24,0x93,0x00, // '['
    0x80,0x81,0x01,0x02,0x02,0x04,0x04,0x08,0x08, // '\'
    0xC9,0x24,0x92,0x4B,0x00, // ']'
    0x30,0x92,0x10, // '^'
    0xFE, // '_'
    0x42, // '`'
    0x72,0x20,0x9E,0x8A,0x27,0x80, // 'a'
    0x82,0x08,0x3C,0x8A,0x28,0xA2,0x8B,0xC0, // 'b'
    0x73,0x28,0x20,0x83,0x07,0x80, // 'c'
    0x08,0x20,0x9E,0x8A,0x28,0xA2,0x89,0xE0, // 'd'
    0x73,0x28,0xBE,0x82,0x27,0x00, // 'e'
    0x18,0x82,0x3E,0x20,0x82,0x08,0x20,0x80, // 'f'
    0x7A,0x28,0xA2,0x8A,0x27,0x82,0x48,0xC0, // 'g'
    0x82,0x08,0x2C,0xCA,0x28,0xA2,0x8A,0x20, // 'h'
    0x20,0x00,0x38,0x20,0x82,0x08,0x23,0xE0, // 'i'
    0x20,0x0E,0x22,0x22,0x22,0x22,0xC0, // 'j'
    0x82,0x08,0x22,0x92,0x8C,0x28,0x92,0x20, // 'k'
    0xE0,0x82,0x08,0x20,0x82,0x08,0x20,0x60, // 'l'
    0xFA,0xAA,0xAA,0xAA,0xAA,0x80, // 'm'
    0xB3,0x28,0xA2,0x8A,0x28,0x80, // 'n'
    0x72,0x28,0xA2,0x8A,0x27,0x00, // 'o'
    0xF2,0x28,0xA2,0x8A,0x2F,0x20,0x82,0x00, // 'p'
    0x7A,0x28,0xA2,0x8A,0x27,0x82,0x08,0x20, // 'q'
    0xF3,0x28,0x20,0x82,0x08,0x00, // 'r'
    0x72,0x28,0x1C,0x0A,0x27,0x00, // 's'
    0x20,0x8F,0x88,0x20,0x82,0x08,0x38, // 't'
    0x8A,0x28,0xA2,0x8A,0x27,0x80, // 'u'
    0x8A,0x25,0x14,0x50,0x82,0x00, // 'v'
    0x82,0x82,0x54,0x54,0x6C,0x28,0x28, // 'w'
    0x89,0x45,0x08,0x51,0x48,0x80, // 'x'
    0x8A,0x25,0x14,0x51,0x82,0x08,0x43,0x00, // 'y'
    0xF8,0x21,0x08,0x42,0x0F,0x80, // 'z'
    0x38,0x82,0x08,0x23,0x02,0x08,0x20,0x83,0x80, // '{'
    0xAA,0xAA,0xAA, // '|'
    0xE0,0x82,0x08,0x20,0x62,0x08,0x20,0x8E,0x00 // '}'
};
const GFXglyph Monospaced_plain_12Glyphs[] PROGMEM = {
// bitmapOffset, width, height, xAdvance, xOffset, yOffset
      {     0,   2,   1,   8,    0,   -1 }, // ' '
      {     1,   2,   9,   8,    3,   -9 }, // '!'
      {     4,   4,   3,   8,    2,   -9 }, // '"'
      {     6,   8,   8,   8,    0,   -8 }, // '#'
      {    14,   6,  11,   8,    1,   -9 }, // '$'
      {    23,   8,   9,   8,    0,   -9 }, // '%'
      {    32,   7,   9,   8,    1,   -9 }, // '&'
      {    40,   2,   3,   8,    3,   -9 }, // '''
      {    41,   4,  11,   8,    3,  -10 }, // '('
      {    47,   4,  11,   8,    2,   -10 }, // ')'
      {    53,   6,   6,   8,    1,    -9 }, // '*'
      {    58,   8,   7,   8,    0,    -7 }, // '+'
      {    65,   3,   3,   8,    2,    -2 }, // ','
      {    67,   4,   1,   8,    2,    -4 }, // '-'
      {    68,   2,   2,   8,    3,    -2 }, // '.'
      {    69,   7,  10,   8,    1,    -9 }, // '/'
      {    78,   7,   9,   8,    1,    -9 }, // '0'
      {    86,   6,   9,   8,    1,    -9 }, // '1'
      {    93,   7,   9,   8,    1,    -9 }, // '2'
      {   101,   7,   9,   8,    1,    -9 }, // '3'
      {   109,   7,   9,   8,    1,    -9 }, // '4'
      {   117,   7,   9,   8,    1,    -9 }, // '5'
      {   125,   7,   9,   8,    1,    -9 }, // '6'
      {   133,   7,   9,   8,    1,    -9 }, // '7'
      {   141,   7,   9,   8,    1,    -9 }, // '8'
      {   149,   7,   9,   8,    1,    -9 }, // '9'
      {   157,   2,   6,   8,    3,    -6 }, // ':'
      {   159,   3,   7,   8,    2,    -6 }, // ';'
      {   162,   7,   6,   8,    1,    -7 }, // '<'
      {   168,   7,   3,   8,    1,    -5 }, // '='
      {   171,   7,   6,   8,    1,    -7 }, // '>'
      {   177,   6,   9,   8,    2,    -9 }, // '?'
      {   184,   7,  10,   8,    1,    -8 }, // '@'
      {   193,   7,   9,   8,    1,    -9 }, // 'A'
      {   201,   7,   9,   8,    1,    -9 }, // 'B'
      {   209,   7,   9,   8,    1,    -9 }, // 'C'
      {   217,   7,   9,   8,    1,    -9 }, // 'D'
      {   225,   7,   9,   8,    1,    -9 }, // 'E'
      {   233,   7,   9,   8,    1,    -9 }, // 'F'
      {   241,   7,   9,   8,    1,    -9 }, // 'G'
      {   249,   7,   9,   8,    1,    -9 }, // 'H'
      {   257,   6,   9,   8,    1,    -9 }, // 'I'
      {   264,   6,   9,   8,    1,    -9 }, // 'J'
      {   271,   7,   9,   8,    1,    -9 }, // 'K'
      {   279,   7,   9,   8,    1,    -9 }, // 'L'
      {   287,   7,   9,   8,    1,    -9 }, // 'M'
      {   295,   7,   9,   8,    1,    -9 }, // 'N'
      {   303,   7,   9,   8,    1,    -9 }, // 'O'
      {   311,   7,   9,   8,    1,    -9 }, // 'P'
      {   319,   7,  11,   8,    1,    -9 }, // 'Q'
      {   329,   8,   9,   8,    1,    -9 }, // 'R'
      {   338,   7,   9,   8,    1,    -9 }, // 'S'
      {   346,   8,   9,   8,    0,    -9 }, // 'T'
      {   355,   7,   9,   8,    1,    -9 }, // 'U'
      {   363,   7,   9,   8,    1,    -9 }, // 'V'
      {   371,   8,   9,   8,    0,    -9 }, // 'W'
      {   380,   7,   9,   8,    1,    -9 }, // 'X'
      {   388,   8,   9,   8,    0,    -9 }, // 'Y'
      {   397,   7,   9,   8,    1,    -9 }, // 'Z'
      {   405,   3,  11,   8,    3,   -10 }, // '['
      {   410,   7,  10,   8,    1,    -9 }, // '\'
      {   419,   3,  11,   8,    2,   -10 }, // ']'
      {   424,   7,   3,   8,    0,    -9 }, // '^'
      {   427,   8,   1,   8,    0,     2 }, // '_'
      {   428,   4,   2,   8,    2,   -10 }, // '`'
      {   429,   6,   7,   8,    1,    -7 }, // 'a'
      {   435,   6,  10,   8,    1,   -10 }, // 'b'
      {   443,   6,   7,   8,    1,    -7 }, // 'c'
      {   449,   6,  10,   8,    1,   -10 }, // 'd'
      {   457,   6,   7,   8,    1,    -7 }, // 'e'
      {   463,   6,  10,   8,    1,   -10 }, // 'f'
      {   471,   6,  10,   8,    1,    -7 }, // 'g'
      {   479,   6,  10,   8,    1,   -10 }, // 'h'
      {   487,   6,  10,   8,    1,   -10 }, // 'i'
      {   495,   4,  13,   8,    2,   -10 }, // 'j'
      {   502,   6,  10,   8,    1,   -10 }, // 'k'
      {   510,   6,  10,   8,    1,   -10 }, // 'l'
      {   518,   6,   7,   8,    1,    -7 }, // 'm'
      {   524,   6,   7,   8,    1,    -7 }, // 'n'
      {   530,   6,   7,   8,    1,    -7 }, // 'o'
      {   536,   6,  10,   8,    1,    -7 }, // 'p'
      {   544,   6,  10,   8,    1,    -7 }, // 'q'
      {   552,   6,   7,   8,    2,    -7 }, // 'r'
      {   558,   6,   7,   8,    1,    -7 }, // 's'
      {   564,   6,   9,   8,    1,    -9 }, // 't'
      {   571,   6,   7,   8,    1,    -7 }, // 'u'
      {   577,   6,   7,   8,    1,    -7 }, // 'v'
      {   583,   8,   7,   8,    0,    -7 }, // 'w'
      {   590,   6,   7,   8,    1,    -7 }, // 'x'
      {   596,   6,  10,   8,    1,    -7 }, // 'y'
      {   604,   6,   7,   8,    1,    -7 }, // 'z'
      {   610,   6,  11,   8,    1,   -10 }, // '{'
      {   619,   2,  12,   8,    3,   -10 }, // '|'
      {   622,   6,  11,   8,    1,   -10 } // '}'
};
const GFXfont Monospaced_plain_12 PROGMEM = {
```

```
(uint8_t  *)Monospaced_plain_12Bitmaps,(GFXglyph *)Monospaced_plain_12Glyphs,0x20, 0x7E, 15};
```

3.12 mono9.h

```
// Created by https://oleddisplay.squix.ch/ Consider a donation
// In case of problems make sure that you are using the font file with the correct version!
const uint8_t Monospaced_plain_9Bitmaps[] PROGMEM = {

    // Bitmap Data:
    0x00, // ' '
    0xAA,0x88, // '!'
    0xAA, // '"'
    0x31,0x4F,0x94,0xF9,0x8A,0x00, // '#'
    0x21,0xEA,0x38,0x38,0xAF,0x08, // '$'
    0xE2,0x8E,0x8C,0x78,0xA3,0x80, // '%'
    0x72,0x10,0xCB,0x51,0xC0, // '&'
    0xA0, // '''
    0x52,0x49,0x22, // '('
    0x91,0x24,0xA4, // ')'
    0xA9,0xC7,0x2A, // '*'
    0x20,0x8F,0x88,0x20, // '+'
    0xA8, // ','
    0xC0, // '-'
    0x80, // '.'
    0x11,0x08,0xC4,0x22,0x00, // '/'
    0x64,0xA5,0x69,0x49,0x80, // '0'
    0xC4,0x44,0x44,0xE0, // '1'
    0x64,0x84,0x66,0x63,0xC0, // '2'
    0x64,0x84,0xC1,0x0B,0x80, // '3'
    0x21,0x18,0xCA,0x78,0x80, // '4'
    0xF4,0x21,0xC1,0x0B,0x80, // '5'
    0x76,0x21,0xE9,0x49,0x80, // '6'
    0xF0,0x88,0x42,0x31,0x00, // '7'
    0x64,0xA4,0xC9,0x49,0x80, // '8'
    0x64,0xA5,0xE1,0x1B,0x80, // '9'
    0x80,0x80, // ':'
    0x80,0xA8, // ';'
    0x09,0xCE,0x06, // '<'
    0xF8,0x0F,0x80, // '='
    0x81,0xC3,0xB0, // '>'
    0xF0,0x88,0x84,0x01,0x00, // '?'
    0x64,0xAD,0x6B,0x61,0x80, // '@'
    0x63,0x19,0x29,0x7A,0x40, // 'A'
    0xE4,0xA5,0xC9,0x4B,0xC0, // 'B'
    0x74,0x21,0x08,0x41,0xC0, // 'C'
    0xE4,0xA5,0x29,0x4B,0x80, // 'D'
    0xF4,0x21,0xE8,0x43,0xC0, // 'E'
    0xF4,0x21,0xE8,0x42,0x00, // 'F'
    0x74,0x21,0x69,0x49,0xC0, // 'G'
    0x94,0xA5,0xE9,0x4A,0x40, // 'H'
    0xE4,0x44,0x44,0xE0, // 'I'
    0x70,0x84,0x21,0x0B,0xC0, // 'J'
    0x95,0x31,0x8A,0x52,0x40, // 'K'
    0x84,0x21,0x08,0x43,0xC0, // 'L'
    0x97,0xBD,0xEF,0x4A,0x40, // 'M'
    0x96,0xB5,0x6B,0x5A,0x40, // 'N'
    0x64,0xA5,0x29,0x49,0x80, // 'O'
    0xE4,0xA5,0xC8,0x42,0x00, // 'P'
    0x64,0xA5,0x29,0x49,0x82, // 'Q'
    0xE2,0x49,0x38,0xB2,0x48,0x80, // 'R'
    0x64,0xA0,0xE1,0x49,0x80, // 'S'
    0xF8,0x82,0x08,0x20,0x82,0x00, // 'T'
    0x94,0xA5,0x29,0x49,0x80, // 'U'
    0x94,0x98,0xC6,0x31,0x80, // 'V'
    0x8A,0x2A,0xB6,0x51,0x45,0x00, // 'W'
    0x94,0x98,0xC6,0x4A,0x40, // 'X'
    0x89,0x45,0x08,0x20,0x82,0x00, // 'Y'
    0xF0,0x88,0x44,0x43,0xC0, // 'Z'
    0xD2,0x49,0x26, // '['
    0x82,0x10,0xC2,0x10,0x40, // '\'
    0xC9,0x24,0x96, // ']'
    0x23,0x40, // '^'
    0xF8, // '_'
    0x88, // '`'
    0xE0,0xBD,0x2F,0x00, // 'a'
    0x84,0x21,0xC9,0x4A,0x5C, // 'b'
    0x74,0x21,0x07,0x00, // 'c'
    0x10,0x84,0xE9,0x4A,0x4E, // 'd'
    0x64,0xBD,0x07,0x00, // 'e'
    0x32,0x11,0xE4,0x21,0x08, // 'f'
    0x74,0xA5,0x27,0x09,0x80, // 'g'
    0x84,0x21,0xE9,0x4A,0x52, // 'h'
    0x20,0x00,0x18,0x20,0x82,0x3E, // 'i'
    0x20,0x06,0x22,0x22,0x2E, // 'j'
    0x84,0x21,0x2A,0x62,0x92, // 'k'
    0xE0,0x82,0x08,0x20,0x82,0x06, // 'l'
    0xFA,0xAA,0xAA,0xA8, // 'm'
    0xF4,0xA5,0x29,0x00, // 'n'
    0x64,0xA5,0x26,0x00, // 'o'
    0xE4,0xA5,0x2E,0x42,0x00, // 'p'
    0x74,0xA5,0x27,0x08,0x40, // 'q'
    0xE8,0x88,0x80, // 'r'
    0xF4,0x1C,0x2F,0x00, // 's'
    0x47,0x90,0x84,0x38, // 't'
    0x94,0xA5,0x2F,0x00, // 'u'
    0x94,0x98,0xC6,0x00, // 'v'
    0x8A,0xA5,0x14,0x50, // 'w'
    0x93,0x18,0xC9,0x00, // 'x'
    0x94,0x98,0xC4,0x23,0x00, // 'y'
    0xF1,0x18,0x8F,0x00, // 'z'
    0x64,0x48,0x44,0x46, // '{'
    0xAA,0xAA,0x80, // '|'
    0xC4,0x42,0x44,0x4C // '}'
};
const GFXglyph Monospaced_plain_9Glyphs[] PROGMEM = {
// bitmapOffset, width, height, xAdvance, xOffset, yOffset
      {      0,    2,    1,    6,     0,    -1 }, // ' '
      {      1,    2,    7,    6,     2,    -7 }, // '!'
      {      3,    4,    2,    6,     1,    -7 }, // '"'
      {      4,    6,    7,    6,     0,    -7 }, // '#'
      {     10,    6,    8,    6,     1,    -7 }, // '$'
      {     16,    6,    7,    6,     0,    -7 }, // '%'
      {     22,    5,    7,    6,     1,    -7 }, // '&'
      {     27,    2,    2,    6,     2,    -7 }, // '''
      {     28,    3,    8,    6,     2,    -8 }, // '('
      {     31,    3,    8,    6,     2,    -8 }, // ')'
      {     34,    6,    4,    6,     0,    -7 }, // '*'
      {     37,    6,    5,    6,     0,    -5 }, // '+'
      {     41,    2,    3,    6,     2,    -1 }, // ','
      {     42,    3,    1,    6,     2,    -3 }, // '-'
      {     43,    2,    1,    6,     2,    -1 }, // '.'
      {     44,    5,    7,    6,     0,    -7 }, // '/'
      {     49,    5,    7,    6,     1,    -7 }, // '0'
      {     54,    4,    7,    6,     2,    -7 }, // '1'
      {     58,    5,    7,    6,     1,    -7 }, // '2'
      {     63,    5,    7,    6,     1,    -7 }, // '3'
      {     68,    5,    7,    6,     1,    -7 }, // '4'
      {     73,    5,    7,    6,     1,    -7 }, // '5'
      {     78,    5,    7,    6,     1,    -7 }, // '6'
      {     83,    5,    7,    6,     1,    -7 }, // '7'
      {     88,    5,    7,    6,     1,    -7 }, // '8'
      {     93,    5,    7,    6,     1,    -7 }, // '9'
      {     98,    2,    5,    6,     2,    -5 }, // ':'
      {    100,    2,    7,    6,     2,    -5 }, // ';'
      {    102,    6,    4,    6,     1,    -5 }, // '<'
      {    105,    6,    3,    6,     0,    -4 }, // '='
      {    108,    6,    4,    6,     1,    -5 }, // '>'
      {    111,    5,    7,    6,     1,    -7 }, // '?'
      {    116,    5,    7,    6,     1,    -6 }, // '@'
      {    121,    5,    7,    6,     1,    -7 }, // 'A'
      {    126,    5,    7,    6,     1,    -7 }, // 'B'
      {    131,    5,    7,    6,     1,    -7 }, // 'C'
      {    136,    5,    7,    6,     1,    -7 }, // 'D'
      {    141,    5,    7,    6,     1,    -7 }, // 'E'
      {    146,    5,    7,    6,     1,    -7 }, // 'F'
      {    151,    5,    7,    6,     1,    -7 }, // 'G'
      {    156,    5,    7,    6,     1,    -7 }, // 'H'
      {    161,    4,    7,    6,     1,    -7 }, // 'I'
      {    165,    5,    7,    6,     1,    -7 }, // 'J'
      {    170,    5,    7,    6,     1,    -7 }, // 'K'
      {    175,    5,    7,    6,     1,    -7 }, // 'L'
      {    180,    5,    7,    6,     1,    -7 }, // 'M'
      {    185,    5,    7,    6,     1,    -7 }, // 'N'
      {    190,    5,    7,    6,     1,    -7 }, // 'O'
      {    195,    5,    7,    6,     1,    -7 }, // 'P'
      {    200,    5,    8,    6,     1,    -7 }, // 'Q'
      {    205,    6,    7,    6,     1,    -7 }, // 'R'
      {    211,    5,    7,    6,     1,    -7 }, // 'S'
      {    216,    6,    7,    6,     0,    -7 }, // 'T'
      {    222,    5,    7,    6,     1,    -7 }, // 'U'
      {    227,    5,    7,    6,     1,    -7 }, // 'V'
      {    232,    6,    7,    6,     0,    -7 }, // 'W'
      {    238,    5,    7,    6,     1,    -7 }, // 'X'
      {    243,    6,    7,    6,     0,    -7 }, // 'Y'
      {    249,    5,    7,    6,     1,    -7 }, // 'Z'
      {    254,    3,    8,    6,     2,    -8 }, // '['
      {    257,    5,    7,    6,     0,    -7 }, // '\'
      {    262,    3,    8,    6,     2,    -8 }, // ']'
      {    265,    6,    2,    6,     0,    -7 }, // '^'
      {    267,    6,    1,    6,     0,     1 }, // '_'
      {   268,   3,   2,   6,    1,   -8 }, // '`'
      {   269,   5,   5,   6,    1,   -5 }, // 'a'
      {   273,   5,   8,   6,    1,   -8 }, // 'b'
      {   278,   5,   5,   6,    1,   -5 }, // 'c'
      {   282,   5,   8,   6,    1,   -8 }, // 'd'
      {   287,   5,   5,   6,    1,   -5 }, // 'e'
      {   291,   5,   8,   6,    1,   -8 }, // 'f'
      {   296,   5,   7,   6,    1,   -5 }, // 'g'
      {   301,   5,   8,   6,    1,   -8 }, // 'h'
      {   306,   6,   8,   6,    0,   -8 }, // 'i'
      {   312,   4,  10,   6,    1,   -8 }, // 'j'
      {   317,   5,   8,   6,    1,   -8 }, // 'k'
      {   322,   6,   8,   6,    0,   -8 }, // 'l'
      {   328,   6,   5,   6,    1,   -5 }, // 'm'
      {   332,   5,   5,   6,    1,   -5 }, // 'n'
      {   336,   5,   5,   6,    1,   -5 }, // 'o'
      {   340,   5,   7,   6,    1,   -5 }, // 'p'
      {   345,   5,   7,   6,    1,   -5 }, // 'q'
      {   350,   4,   5,   6,    2,   -5 }, // 'r'
      {   353,   5,   5,   6,    1,   -5 }, // 's'
      {   357,   5,   6,   6,    1,   -6 }, // 't'
      {   361,   5,   5,   6,    1,   -5 }, // 'u'
      {   365,   5,   5,   6,    1,   -5 }, // 'v'
      {   369,   6,   5,   6,    0,   -5 }, // 'w'
      {   373,   5,   5,   6,    1,   -5 }, // 'x'
      {   377,   5,   7,   6,    1,   -5 }, // 'y'
      {   382,   5,   5,   6,    1,   -5 }, // 'z'
      {   386,   4,   8,   6,    1,   -8 }, // '{'
      {   390,   2,   9,   6,    2,   -8 }, // '|'
      {   393,   4,   8,   6,    1,   -8 } // '}'
};
const GFXfont Monospaced_plain_9 PROGMEM = {
(uint8_t  *)Monospaced_plain_9Bitmaps,(GFXglyph *)Monospaced_plain_9Glyphs,0x20, 0x7E, 12};
```

3.13 opensans12.h

```
// Created by https://oleddisplay.squix.ch/ Consider a donation
// In case of problems make sure that you are using the font file with the correct version!
const uint8_t Open_Sans_Regular_12Bitmaps[] PROGMEM = {

    // Bitmap Data:
    0x00, // ' '
    0xAA,0xA2,0x80, // '!'
    0xAA,0xA0, // '"'
    0x12,0x0A,0x05,0x0F,0xE2,0x41,0x23,0xF8,0x50,0x48,0x00, // '#'
    0x21,0xEA,0x28,0x60,0xE2,0x8A,0xF0,0x80, // '$'
    0xC2,0x52,0x29,0x15,0xEA,0xD7,0xA8,0x94,0x4A,0x47,0x00, // '%'
    0x70,0x24,0x12,0x0A,0x06,0x04,0x92,0x29,0x08,0x7A,0x00, // '&'
    0xA8, // "'
    0x4C,0x88,0x88,0x88,0x8C,0x40, // '('
    0x46,0x22,0x22,0x22,0x26,0x40, // ')'
    0x20,0x8F,0x94,0xD8, // '*'
    0x20,0x82,0x3E,0x20,0x82,0x00, // '+'
    0x4A,0x00, // ','
    0xC0, // '-'
    0xA0, // '.'
    0x11,0x08,0x44,0x21,0x08,0x80, // '/'
    0x72,0x28,0xA2,0x8A,0x28,0xA2,0x70, // '0'
    0x26,0xA2,0x22,0x22,0x20, // '1'
    0x72,0x20,0x82,0x10,0x84,0x20,0xF8, // '2'
    0xF0,0x20,0x86,0x70,0x20,0x82,0xF0, // '3'
    0x08,0x18,0x18,0x28,0x48,0x48,0xFE,0x08,0x08, // '4'
    0xFA,0x08,0x3C,0x08,0x20,0x82,0xF0, // '5'
    0x39,0x08,0x3C,0x8A,0x28,0xA2,0x70, // '6'
    0xF8,0x20,0x84,0x10,0x82,0x10,0x40, // '7'
    0x72,0x28,0xB6,0x72,0x28,0xA2,0xF0, // '8'
    0x72,0x28,0xA2,0x89,0xE0,0x86,0xE0, // '9'
    0xA0,0xA0, // ':'
    0x48,0x00,0x94, // ';'
    0x08,0xCC,0x30,0x30,0x20, // '<'
    0xF8,0x0F,0x80, // '='
    0x81,0x81,0x86,0x62,0x00, // '>'
    0xF2,0x20,0x84,0x20,0x80,0x18,0x60, // '?'
    0x3E,0x10,0x49,0xEA,0x4A,0xA2,0xA8,0xAA,0x4A,0x6C,0x40,0x0F,0x80, // '@'
    0x18,0x0C,0x06,0x04,0x82,0x43,0xF1,0x08,0x84,0x81,0x00, // 'A'
    0xF9,0x0A,0x14,0x2F,0x90,0xA1,0x42,0xF8, // 'B'
    0x3E,0x40,0x80,0x80,0x80,0x80,0x80,0x40,0x3C, // 'C'
    0xF8,0x84,0x82,0x82,0x82,0x82,0x82,0x84,0xF8, // 'D'
    0xFA,0x08,0x20,0xFA,0x08,0x20,0xF8, // 'E'
    0xFA,0x08,0x20,0xFA,0x08,0x20,0x80, // 'F'
    0x3E,0x40,0x80,0x80,0x8E,0x82,0x82,0x42,0x3E, // 'G'
    0x82,0x82,0x82,0x82,0xFE,0x82,0x82,0x82,0x82, // 'H'
    0xAA,0xAA,0x80, // 'I'
    0x22,0x22,0x22,0x22,0x22,0xC0, // ’J’
    0x89,0x12,0x45,0x0E,0x12,0x24,0x44,0x84, // ’K’
    0x82,0x08,0x20,0x82,0x08,0x20,0xF8, // ’L’
    0xC1,0xB0,0x6A,0x2A,0x8A,0xA2,0xA5,0x29,0x4A,0x22,0x88,0x80, // ’M’
    0xC2,0xC2,0xA2,0xA2,0x92,0x8A,0x8A,0x86,0x86, // ’N’
    0x78,0xC6,0x82,0x82,0x82,0x82,0x82,0xC6,0x78, // ’O’
    0xF2,0x28,0xA2,0x8B,0xC8,0x20,0x80, // ’P’
    0x78,0xC6,0x82,0x82,0x82,0x82,0x82,0xC6,0x7C,0x08,0x04, // ’Q’
    0xF1,0x12,0x24,0x4F,0x12,0x26,0x44,0x8C, // ’R’
    0x7A,0x08,0x30,0x70,0x20,0x82,0xF0, // ’S’
    0xFE,0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x10, // ’T’
    0x82,0x82,0x82,0x82,0x82,0x82,0x82,0x44,0x78, // ’U’
    0x82,0x44,0x44,0x44,0x28,0x28,0x28,0x28,0x10, // ’V’
    0x84,0x24,0xC4,0x4A,0x44,0xA4,0x4A,0x43,0x18,0x31,0x83,0x18,0x31,0x80, // ’W’
    0x44,0x44,0x28,0x28,0x10,0x28,0x28,0x44,0x82, // ’X’
    0x82,0x44,0x44,0x28,0x38,0x10,0x10,0x10,0x10, // ’Y’
    0x7C,0x04,0x08,0x08,0x10,0x20,0x20,0x40,0xFE, // ’Z’
    0xE8,0x88,0x88,0x88,0x88,0xE0, // ’[’
    0x82,0x10,0x82,0x10,0x84,0x10, // ’\’
    0xE2,0x22,0x22,0x22,0x22,0xE0, // ’]’
    0x10,0x30,0x28,0x48,0x44,0x44, // ’^’
    0xF8, // ’_’
    0x44, // ’‘’
    0x70,0x27,0xA2,0x8B,0xE0, // ’a’
    0x82,0x08,0x3C,0x8A,0x28,0xA2,0xF0, // ’b’
    0x7A,0x08,0x20,0x81,0xC0, // ’c’
    0x08,0x20,0x9E,0x8A,0x28,0xA2,0x78, // ’d’
    0x72,0x2F,0xA0,0x81,0xE0, // ’e’
    0x31,0x04,0x3C,0x41,0x04,0x10,0x40, // ’f’
    0x3E,0x44,0x44,0x78,0x40,0x3C,0xC2,0x84,0x7C, // ’g’
    0x82,0x08,0x3E,0x8A,0x28,0xA2,0x88, // ’h’
    0x8A,0xAA, // ’i’
    0x20,0x22,0x22,0x22,0x22,0xE0, // ’j’
    0x82,0x08,0x24,0xA3,0x0A,0x24,0x90, // ’k’
    0xAA,0xAA,0x80, // ’l’
    0xF7,0x22,0x28,0x8A,0x22,0x88,0xA2,0x20, // ’m’
    0xFA,0x28,0xA2,0x8A,0x20, // ’n’
    0x72,0x28,0xA2,0x89,0xC0, // ’o’
    0xF2,0x28,0xA2,0x8B,0xC8,0x20,0x80, // ’p’
    0x7A,0x28,0xA2,0x89,0xE0,0x82,0x08, // ’q’
    0xB6,0x21,0x08,0x40, // ’r’
    0xF4,0x30,0x61,0x78, // ’s’
    0x47,0x90,0x84,0x21,0xC0, // ’t’
    0x8A,0x28,0xA2,0x8B,0xE0, // ’u’
    0x84,0x91,0x22,0x43,0x06,0x00, // ’v’
    0x88,0x95,0x45,0x51,0x54,0x65,0x08,0x80, // ’w’
    0x48,0x90,0xC1,0x84,0x99,0x00, // ’x’
    0x84,0x91,0x22,0x43,0x06,0x08,0x10,0xC0, // ’y’
    0xF0,0x42,0x10,0x83,0xE0, // ’z’
    0x18,0x82,0x08,0x23,0x02,0x08,0x20,0x81,0x80, // ’{’
    0xAA,0xAA,0xAA, // ’|’
    0xC0,0x82,0x08,0x20,0x62,0x08,0x20,0x8C,0x00 // ’}’
};
const GFXglyph Open_Sans_Regular_12Glyphs[] PROGMEM = {
// bitmapOffset, width, height, xAdvance, xOffset, yOffset
      {     0,   2,   1,   4,    0,   -1 }, // ’ ’
      {     1,   2,   9,   4,    1,   -9 }, // ’!’
      {     4,   4,   3,   6,    1,   -9 }, // ’"’
      {     6,   9,   9,   9,    0,   -9 }, // ’#’
      {    17,   6,  10,   8,    1,   -9 }, // ’$’
      {    25,   9,   9,  11,    1,   -9 }, // ’%’
      {    36,   9,   9,  10,    1,   -9 }, // ’&’
      {    47,   2,   3,   4,    1,   -9 }, // ”’
      {    48,   4,  11,   5,    1,   -9 }, // ’(’
      {    54,   4,  11,   5,    0,   -9 }, // ’)’
      {    60,   6,   5,   8,    1,   -9 }, // ’*’
      {    64,   6,   7,   8,    1,   -8 }, // ’+’
      {    70,   3,   3,   4,    0,   -1 }, // ’,’
      {    72,   3,   1,   5,    1,   -4 }, // ’-’
      {    73,   2,   2,   4,    1,   -2 }, // ’.’
      {    74,   5,   9,   5,    0,   -9 }, // ’/’
      {    80,   6,   9,   8,    1,   -9 }, // ’0’
      {    87,   4,   9,   8,    1,   -9 }, // ’1’
      {    92,   6,   9,   8,    1,   -9 }, // ’2’
      {    99,   6,   9,   8,    1,   -9 }, // ’3’
      {   106,   8,   9,   8,    0,   -9 }, // ’4’
      {   115,   6,   9,   8,    1,   -9 }, // ’5’
      {   122,   6,   9,   8,    1,   -9 }, // ’6’
      {   129,   6,   9,   8,    1,   -9 }, // ’7’
      {   136,   6,   9,   8,    1,   -9 }, // ’8’
      {   143,   6,   9,   8,    1,   -9 }, // ’9’
      {   150,   2,   6,   4,    1,   -6 }, // ’:’
      {   152,   3,   8,   4,    0,   -6 }, // ’;’
      {   155,   6,   6,   8,    1,   -7 }, // ’<’
      {   160,   6,   3,   8,    1,   -6 }, // ’=’
      {   163,   6,   6,   8,    1,   -7 }, // ’>’
      {   168,   6,   9,   6,    0,   -9 }, // ’?’
      {   175,  10,  10,  12,    1,   -9 }, // '@'
      {   188,   9,   9,   9,    0,   -9 }, // 'A'
      {   199,   7,   9,   9,    1,   -9 }, // 'B'
      {   207,   8,   9,   9,    1,   -9 }, // 'C'
      {   216,   8,   9,  10,    1,   -9 }, // 'D'
      {   225,   6,   9,   8,    1,   -9 }, // 'E'
      {   232,   6,   9,   7,    1,   -9 }, // 'F'
      {   239,   8,   9,  10,    1,   -9 }, // 'G'
      {   248,   8,   9,  10,    1,   -9 }, // 'H'
      {   257,   2,   9,   4,    1,   -9 }, // 'I'
      {   260,   4,  11,   4,   -1,   -9 }, // 'J'
      {   266,   7,   9,   8,    1,   -9 }, // 'K'
      {   274,   6,   9,   7,    1,   -9 }, // 'L'
      {   281,  10,   9,  12,    1,   -9 }, // 'M'
      {   293,   8,   9,  10,    1,   -9 }, // 'N'
      {   302,   8,   9,  10,    1,   -9 }, // 'O'
      {   311,   6,   9,   8,    1,   -9 }, // 'P'
      {   318,   8,  11,  10,    1,   -9 }, // 'Q'
      {   329,   7,   9,   8,    1,   -9 }, // 'R'
      {   337,   6,   9,   8,    1,   -9 }, // 'S'
      {   344,   8,   9,   8,    0,   -9 }, // 'T'
      {   353,   8,   9,  10,    1,   -9 }, // 'U'
      {   362,   8,   9,   8,    0,   -9 }, // 'V'
      {   371,  12,   9,  12,    0,   -9 }, // 'W'
      {   385,   8,   9,   8,    0,   -9 }, // 'X'
      {   394,   8,   9,   8,    0,   -9 }, // 'Y'
      {   403,   8,   9,   8,    0,   -9 }, // 'Z'
      {   412,   4,  11,   5,    1,   -9 }, // '['
      {   418,   5,   9,   5,    0,   -9 }, // '\'
      {   424,   4,  11,   5,    0,   -9 }, // ']'
      {   430,   8,   6,   8,    0,   -9 }, // '^'
      {   436,   6,   1,   6,    0,    1 }, // '_'
      {   437,   4,   2,   8,    2,   -9 }, // '`'
      {   438,   6,   6,   8,    1,   -6 }, // 'a'
      {   443,   6,   9,   8,    1,   -9 }, // 'b'
      {   450,   6,   6,   7,    1,   -6 }, // 'c'
      {   455,   6,   9,   8,    1,   -9 }, // 'd'
      {   462,   6,   6,   8,    1,   -6 }, // 'e'
      {   467,   6,   9,   5,    0,   -9 }, // 'f'
      {   474,   8,   9,   8,    0,   -6 }, // 'g'
      {   483,   6,   9,   8,    1,   -9 }, // 'h'
      {   490,   2,   8,   4,    1,   -8 }, // 'i'
      {   492,   4,  11,   4,   -1,   -8 }, // 'j'
      {   498,   6,   9,   7,    1,   -9 }, // 'k'
      {   505,   2,   9,   4,    1,   -9 }, // 'l'
      {   508,  10,   6,  12,    1,   -6 }, // 'm'
      {   516,   6,   6,   8,    1,   -6 }, // 'n'
      {   521,   6,   6,   8,    1,   -6 }, // 'o'
      {   526,   6,   9,   8,    1,   -6 }, // 'p'
      {   533,   6,   9,   8,    1,   -6 }, // 'q'
      {   540,   5,   6,   6,    1,   -6 }, // 'r'
      {   544,   5,   6,   7,    1,   -6 }, // 's'
      {   548,   5,   7,   5,    0,   -7 }, // 't'
      {   553,   6,   6,   8,    1,   -6 }, // 'u'
      {   558,   7,   6,   7,    0,   -6 }, // 'v'
      {   564,  10,   6,  10,    0,   -6 }, // 'w'
      {   572,   7,   6,   7,    0,   -6 }, // 'x'
      {   578,   7,   9,   7,    0,   -6 }, // 'y'
      {   586,   6,   6,   7,    1,   -6 }, // 'z'
      {   591,   6,  11,   6,    0,   -9 }, // '{'
      {   600,   2,  12,   8,    3,   -9 }, // '|'
      {   603,   6,  11,   6,    0,   -9 } // '}'
};
const GFXfont Open_Sans_Regular_12 PROGMEM = {
(uint8_t  *)Open_Sans_Regular_12Bitmaps,(GFXglyph *)Open_Sans_Regular_12Glyphs,0x20, 0x7E, 17};
```

3.14 opensans9.h

```
// Created by https://oleddisplay.squix.ch/ Consider a donation
// In case of problems make sure that you are using the font file with the correct version!
const uint8_t Open_Sans_Regular_9Bitmaps[] PROGMEM = {

    // Bitmap Data:
    0x00, // ' '
    0xAA,0x88, // '!'
    0xD8, // '"'
    0x28,0x51,0xF2,0x8F,0x8A,0x14,0x00, // '#'
    0x4E,0xCC,0x66,0xE4, // '$'
    0x68,0xA8,0xB0,0x7C,0x1A,0x2A,0x2C, // '%'
    0x61,0x22,0x82,0x0A,0x53,0x3E,0x00, // '&'
    0xA0, // '''
    0x44,0x88,0x88,0x44, // '('
    0x44,0x22,0x22,0x44, // ')'
    0x23,0xE2,0x14, // '*'
    0x20,0x8F,0x88,0x20, // '+'
    0xA0, // ','
    0xE0, // '-'
    0x80, // '.'
    0x22,0x44,0x48,0x80, // '/'
    0x71,0x48,0xA2,0x89,0x47,0x00, // '0'
    0x59,0x24,0x90, // '1'
    0x70,0x41,0x04,0x21,0x0F,0x80, // '2'
    0x70,0x84,0xC1,0x0B,0xC0, // '3'
    0x10,0xC3,0x14,0x93,0xE1,0x00, // '4'
    0xE4,0x21,0xC1,0x0B,0x80, // '5'
    0x31,0x04,0x1C,0x89,0x27,0x00, // '6'
    0xF8,0x41,0x04,0x20,0x84,0x00, // '7'
    0x71,0x45,0x08,0x52,0x27,0x00, // '8'
    0x72,0x48,0x9E,0x08,0x46,0x00, // '9'
    0x80,0x80, // ':'
    0x40,0x09,0x00, // ';'
    0x19,0x86,0x06, // '<'
    0xE0,0xE0, // '='
    0xC0,0xC3,0x30, // '>'
    0xE0,0x84,0x44,0x01,0x00, // '?'
    0x79,0x0A,0xF6,0x6C,0xD7,0xA0,0x3C, // '@'
    0x30,0x60,0xC2,0x47,0x91,0x21,0x00, // 'A'
    0xF4,0xA5,0xC9,0x4B,0xC0, // 'B'
    0x7A,0x08,0x20,0x82,0x07,0x00, // 'C'
    0xF2,0x28,0xA2,0x8A,0x2F,0x00, // 'D'
    0xE8,0x8E,0x88,0xE0, // 'E'
    0xF4,0x21,0xE8,0x42,0x00, // 'F'
    0x7A,0x08,0x26,0x8A,0x27,0x80, // 'G'
    0x8A,0x28,0xBE,0x8A,0x28,0x80, // 'H'
    0xAA,0xA8, // 'I'
    0x49,0x24,0x92,0xC0, // 'J'
    0x92,0x4A,0x30,0xA2,0x48,0x80, // 'K'
    0x84,0x21,0x08,0x43,0xC0, // 'L'
    0x85,0x9B,0x36,0x6B,0x56,0xA9,0x00, // 'M'
    0x8B,0x2C,0xAA,0x9A,0x68,0x80, // 'N'
    0x72,0x28,0xA2,0x8A,0x27,0x00, // 'O'
    0xE4,0xA5,0x4E,0x42,0x00, // 'P'
    0x72,0x28,0xA2,0x8A,0x27,0x04, // 'Q'
    0xF2,0x49,0x38,0xA2,0x48,0x80, // 'R'
    0x72,0x04,0x0C,0x08,0x2F,0x00, // 'S'
    0xF8,0x82,0x08,0x20,0x82,0x00, // 'T'
    0x8A,0x28,0xA2,0x8A,0x27,0x00, // 'U'
    0x8A,0x25,0x14,0x50,0x82,0x00, // 'V'
    0x91,0x4C,0x96,0x8B,0x46,0x63,0x30,0x90, // 'W'
    0x89,0x45,0x08,0x51,0x48,0x80, // 'X'
    0x89,0x45,0x08,0x20,0x82,0x00, // 'Y'
    0xF8,0x41,0x08,0x41,0x0F,0x80, // 'Z'
    0xD2,0x49,0x26, // '['
    0x88,0x44,0x42,0x20, // '\'
    0xC9,0x24,0x96, // ']'
    0x21,0x45,0x22, // '^'
    0xF0, // '_'
    0xA0, // '`'
    0x70,0x9D,0x27,0x00, // 'a'
    0x84,0x3D,0x29,0x4B,0xC0, // 'b'
    0x72,0x20,0x87,0x00, // 'c'
    0x10,0xBD,0x29,0x4B,0xC0, // 'd'
    0x71,0x2F,0x90,0x70, // 'e'
    0x64,0xE4,0x44,0x40, // 'f'
    0x7A,0x47,0x10,0x72,0x2F,0x00, // 'g'
    0x84,0x3D,0x29,0x4A,0x40, // 'h'
    0x8A,0xA8, // 'i'
    0x41,0x24,0x92,0xC0, // 'j'
    0x84,0x29,0x8C,0x52,0x40, // 'k'
    0xAA,0xA8, // 'l'
    0xED,0x2A,0x54,0xA9,0x40, // 'm'
    0xF4,0xA5,0x29,0x00, // 'n'
    0x71,0x48,0x94,0x70, // 'o'
    0xF4,0xA5,0x2F,0x42,0x00, // 'p'
    0xF4,0xA5,0x2F,0x08,0x40, // 'q'
    0xE8,0x88,0x80, // 'r'
    0x64,0x18,0x2E,0x00, // 's'
    0x4E,0x44,0x46, // 't'
    0x94,0xA5,0x2F,0x00, // 'u'
    0x89,0x45,0x14,0x20, // 'v'
    0x92,0xAA,0x6C,0x6C,0x44, // 'w'
    0x51,0x42,0x14,0x48, // 'x'
    0x89,0x45,0x14,0x20,0x8C,0x00, // 'y'
    0xF1,0x10,0x8F,0x00, // 'z'
    0x64,0x44,0x44,0x46, // '{'
    0xAA,0xAA,0x80, // '|'
    0xC4,0x44,0x44,0x4C // '}'
};
const GFXglyph Open_Sans_Regular_9Glyphs[] PROGMEM = {
// bitmapOffset, width, height, xAdvance, xOffset, yOffset
```

```
    {     0,    2,    1,    3,     0,    -1 }, // ' '
    {     1,    2,    7,    3,     1,    -7 }, // '!'
    {     3,    3,    2,    5,     1,    -7 }, // '"'
    {     4,    7,    7,    7,     0,    -7 }, // '#'
    {    11,    4,    8,    6,     1,    -7 }, // '$'
    {    15,    8,    7,    8,     0,    -7 }, // '%'
    {    22,    7,    7,    8,     1,    -7 }, // '&'
    {    29,    2,    2,    3,     1,    -7 }, // ''
    {    30,    4,    8,    4,     0,    -7 }, // '('
    {    34,    4,    8,    4,     0,    -7 }, // ')'
    {    38,    6,    4,    6,     0,    -7 }, // '*'
    {    41,    6,    5,    6,     0,    -6 }, // '+'
    {    45,    2,    2,    3,     0,    -1 }, // ','
    {    46,    4,    1,    4,     0,    -3 }, // '-'
    {    47,    2,    1,    3,     1,    -1 }, // '.'
    {    48,    4,    7,    4,     0,    -7 }, // '/'
    {    52,    6,    7,    6,     0,    -7 }, // '0'
    {    58,    3,    7,    6,     1,    -7 }, // '1'
    {    61,    6,    7,    6,     0,    -7 }, // '2'
    {    67,    5,    7,    6,     0,    -7 }, // '3'
    {    72,    6,    7,    6,     0,    -7 }, // '4'
    {    78,    5,    7,    6,     1,    -7 }, // '5'
    {    83,    6,    7,    6,     0,    -7 }, // '6'
    {    89,    6,    7,    6,     0,    -7 }, // '7'
    {    95,    6,    7,    6,     0,    -7 }, // '8'
    {   101,    6,    7,    6,     0,    -7 }, // '9'
    {   107,    2,    5,    3,     1,    -5 }, // ':'
    {   109,    3,    6,    3,     0,    -5 }, // ';'
    {   112,    6,    4,    6,     0,    -5 }, // '<'
    {   115,    4,    3,    6,     1,    -5 }, // '='
    {   117,    6,    4,    6,     0,    -5 }, // '>'
    {   120,    5,    7,    5,     0,    -7 }, // '?'
    {   125,    7,    8,    9,     1,    -7 }, // '@'
    {   132,    7,    7,    7,     0,    -7 }, // 'A'
    {   139,    5,    7,    7,     1,    -7 }, // 'B'
    {   144,    6,    7,    7,     1,    -7 }, // 'C'
    {   150,    6,    7,    8,     1,    -7 }, // 'D'
    {   156,    4,    7,    6,     1,    -7 }, // 'E'
    {   160,    5,    7,    6,     1,    -7 }, // 'F'
    {   165,    6,    7,    8,     1,    -7 }, // 'G'
    {   171,    6,    7,    8,     1,    -7 }, // 'H'
    {   177,    2,    7,    4,     1,    -7 }, // 'I'
    {   179,    3,    9,    3,    -1,    -7 }, // 'J'
    {   183,    6,    7,    7,     1,    -7 }, // 'K'
    {   189,    5,    7,    6,     1,    -7 }, // 'L'
    {   194,    7,    7,    9,     1,    -7 }, // 'M'
    {   201,    6,    7,    8,     1,    -7 }, // 'N'
    {   207,    6,    7,    8,     1,    -7 }, // 'O'
    {   213,    5,    7,    6,     1,    -7 }, // 'P'
    {   218,    6,    8,    8,     1,    -7 }, // 'Q'
    {   224,    6,    7,    7,     1,    -7 }, // 'R'
    {   230,    6,    7,    6,     0,    -7 }, // 'S'
    {   236,    6,    7,    6,     0,    -7 }, // 'T'
    {   242,    6,    7,    8,     1,    -7 }, // 'U'
    {   248,    6,    7,    6,     0,    -7 }, // 'V'
    {   254,    9,    7,    9,     0,    -7 }, // 'W'
    {   262,    6,    7,    6,     0,    -7 }, // 'X'
    {   268,    6,    7,    6,     0,    -7 }, // 'Y'
    {   274,    6,    7,    6,     0,    -7 }, // 'Z'
    {   280,    3,    8,    4,     1,    -7 }, // '['
    {   283,    4,    7,    4,     0,    -7 }, // '\'
    {   287,    3,    8,    4,     0,    -7 }, // ']'
    {   290,    6,    4,    6,     0,    -6 }, // '^'
    {   293,    5,    1,    5,     0,     1 }, // '_'
    {   294,    2,    2,    6,     2,    -7 }, // '`'
    {   295,    5,    5,    6,     0,    -5 }, // 'a'
    {   299,    5,    7,    7,     1,    -7 }, // 'b'
    {   304,    5,    5,    5,     0,    -5 }, // 'c'
    {   308,    5,    7,    7,     1,    -7 }, // 'd'
    {   313,    6,    5,    6,     0,    -5 }, // 'e'
    {   317,    4,    7,    4,     0,    -7 }, // 'f'
    {   321,    6,    7,    6,     0,    -5 }, // 'g'
    {   327,    5,    7,    7,     1,    -7 }, // 'h'
    {   332,    2,    7,    3,     1,    -7 }, // 'i'
    {   334,    3,    9,    3,    -1,    -7 }, // 'j'
    {   338,    5,    7,    6,     1,    -7 }, // 'k'
    {   343,    2,    7,    3,     1,    -7 }, // 'l'
    {   345,    7,    5,    9,     1,    -5 }, // 'm'
    {   350,    5,    5,    7,     1,    -5 }, // 'n'
    {   354,    6,    5,    6,     0,    -5 }, // 'o'
    {   358,    5,    7,    7,     1,    -5 }, // 'p'
    {   363,    5,    7,    7,     1,    -5 }, // 'q'
    {   368,    4,    5,    5,     1,    -5 }, // 'r'
    {   371,    5,    5,    5,     0,    -5 }, // 's'
    {   375,    4,    6,    4,     0,    -6 }, // 't'
    {   378,    5,    5,    7,     1,    -5 }, // 'u'
    {   382,    6,    5,    6,     0,    -5 }, // 'v'
    {   386,   8,   5,   8,    0,   -5 }, // 'w'
    {   391,   6,   5,   6,    0,   -5 }, // 'x'
    {   395,   6,   7,   6,    0,   -5 }, // 'y'
    {   401,   5,   5,   5,    0,   -5 }, // 'z'
    {   405,   4,   8,   4,    0,   -7 }, // '{'
    {   409,   2,   9,   6,    2,   -7 }, // '|'
    {   412,   4,   8,   4,    0,   -7 } // '}'
};
const GFXfont Open_Sans_Regular_9 PROGMEM = {
(uint8_t  *)Open_Sans_Regular_9Bitmaps,(GFXglyph *)Open_Sans_Regular_9Glyphs,0x20, 0x7E, 13};
```

3.15 roboto12.h

```
// Created by https://oleddisplay.squix.ch/ Consider a donation
// In case of problems make sure that you are using the font file with the correct version!
const uint8_t Roboto_Condensed_Light_12Bitmaps[] PROGMEM = {

    // Bitmap Data:
    0x00, // ' '
    0xAA,0xA8,0x80, // '!'
    0xDB,0x00, // '"'
    0x51,0x4F,0x94,0x53,0xEA,0x28,0xA0, // '#'
    0x43,0x25,0x28,0x30,0x52,0x93,0x90, // '$'
    0xC1,0x52,0xC6,0x82,0x05,0x95,0x2A,0x0C, // '%'
    0x30,0xA1,0x43,0x06,0x12,0xA6,0x44,0x74, // '&'
    0xA8, // '''
    0x4A,0x49,0x24,0x92,0x24, // '('
    0x84,0x42,0x22,0x22,0x22,0x44,0x80, // ')'
    0x22,0xA7,0x14,0x90, // '*'
    0x20,0x82,0x3E,0x20,0x82,0x00, // '+'
    0x4A,0x00, // ','
    0xE0, // '-'
    0x80, // '.'
    0x11,0x08,0x44,0x21,0x08,0x84,0x00, // '/'
    0x64,0xA5,0x29,0x4A,0x52,0x60, // '0'
    0x6A,0x22,0x22,0x22,0x20, // '1'
    0x64,0xA4,0x22,0x11,0x10,0xF0, // '2'
    0x64,0x84,0x26,0x08,0x52,0x60, // '3'
    0x10,0x20,0xC1,0x85,0x12,0x3F,0x08,0x10, // '4'
    0x74,0x21,0xE9,0x08,0x52,0x70, // '5'
    0x62,0x21,0xC9,0x4A,0x52,0x60, // '6'
    0xF8,0x20,0x84,0x10,0x82,0x08,0x40, // '7'
    0x64,0xA5,0x26,0x4A,0x52,0x70, // '8'
    0x64,0xA5,0x29,0x38,0x42,0x60, // '9'
    0x80,0x20, // ':'
    0x40,0x00,0x94, // ';'
    0x11,0x91,0x83,0x08, // '<'
    0xF0,0x3C, // '='
    0x82,0x0C,0x64,0x40, // '>'
    0xEA,0x22,0x24,0x40,0x40, // '?'
    0x1E,0x08,0x44,0xD2,0x54,0xA4,0xA9,0x4A,0x52,0x6C,0x40,0x18,0x03,0xC0, // '@'
    0x10,0x30,0x28,0x28,0x28,0x48,0x7C,0x44,0x86, // 'A'
    0xF2,0x49,0x24,0xE2,0x48,0xA4,0xF0, // 'B'
    0x72,0x28,0xA0,0x82,0x08,0xA2,0x70, // 'C'
    0xF2,0x28,0xA2,0x8A,0x28,0xA2,0xF0, // 'D'
    0xF4,0x21,0x0F,0x42,0x10,0xF0, // 'E'
    0xF4,0x21,0x0F,0x42,0x10,0x80, // 'F'
    0x72,0x28,0xA0,0x82,0x68,0xA2,0x78, // 'G'
    0x8A,0x28,0xA2,0xFA,0x28,0xA2,0x88, // 'H'
    0xAA,0xAA,0x80, // 'I'
    0x10,0x84,0x21,0x0A,0x52,0xE0, // 'J'
    0x89,0x22,0x85,0x0E,0x14,0x24,0x44,0x8C, // 'K'
    0x84,0x21,0x08,0x42,0x10,0xF0, // 'L'
    0x82,0x82,0xC6,0xC6,0xCA,0xAA,0xAA,0xB2,0x92, // 'M'
    0x8A,0x2C,0xB2,0xAA,0xA9,0xA2,0x88, // 'N'
    0x72,0x28,0xA2,0x8A,0x28,0xA2,0x70, // 'O'
    0xF2,0x48,0xA2,0x93,0xC8,0x20,0x80, // 'P'
    0x72,0x28,0xA2,0x8A,0x28,0xA2,0x70,0x20, // 'Q'
    0xF2,0x28,0xA2,0x8B,0xC9,0x24,0x88, // 'R'
    0x78,0x92,0x12,0x03,0x01,0x21,0x24,0x78, // 'S'
    0xFC,0x40,0x81,0x02,0x04,0x08,0x10,0x20, // 'T'
    0x8A,0x28,0xA2,0x8A,0x28,0xA2,0x70, // 'U'
    0x85,0x09,0x22,0x44,0x85,0x0C,0x18,0x10, // 'V'
    0x88,0xA2,0x25,0x51,0x54,0x55,0x15,0x46,0x50,0x88,0x22,0x00, // 'W'
    0xC4,0x90,0xA1,0x81,0x06,0x12,0x24,0x84, // 'X'
    0x84,0x91,0x21,0x83,0x02,0x08,0x10,0x20, // 'Y'
    0xF8,0x41,0x08,0x41,0x08,0x20,0xF8, // 'Z'
    0xD2,0x49,0x24,0x92,0x60, // '['
    0x84,0x10,0x84,0x10,0x84,0x10,0x80, // '\'
    0xC9,0x24,0x92,0x49,0x60, // ']'
    0x43,0x29,0x20, // '^'
    0xF8, // '_'
    0x88, // '`'
    0x64,0x84,0xE9,0x4B,0xC0, // 'a'
    0x84,0x3D,0x29,0x4A,0x52,0xF0, // 'b'
    0x72,0x28,0x20,0x82,0x27,0x00, // 'c'
    0x10,0xBD,0x29,0x4A,0x52,0xF0, // 'd'
    0x72,0x48,0xBE,0x82,0x07,0x80, // 'e'
    0x68,0xC8,0x88,0x88,0x80, // 'f'
    0xF4,0xA5,0x29,0x4B,0xD2,0x60, // 'g'
    0x84,0x3D,0x29,0x4A,0x52,0x90, // 'h'
    0x8A,0xAA,0x80, // 'i'
    0x41,0x24,0x92,0x4A,0x00, // 'j'
    0x84,0x2D,0x4C,0x62,0x94,0x90, // 'k'
    0xAA,0xAA,0x80, // 'l'
    0xEE,0x92,0x92,0x92,0x92,0x92,0x92, // 'm'
    0xF4,0xA5,0x29,0x4A,0x40, // 'n'
    0x64,0xA5,0x29,0x49,0x80, // 'o'
    0xF4,0xA5,0x29,0x4B,0xD0,0x80, // 'p'
    0xF4,0xA5,0x29,0x4B,0xC2,0x10, // 'q'
    0xE8,0x88,0x88,0x80, // 'r'
    0x72,0x48,0x18,0x12,0x67,0x00, // 's'
    0x44,0xE4,0x44,0x44,0x60, // 't'
    0x94,0xA5,0x29,0x49,0xC0, // 'u'
    0x8A,0x45,0x14,0x61,0x82,0x00, // 'v'
    0x92,0x9A,0x5A,0x6A,0x6C,0x2C,0x24, // 'w'
    0x99,0x46,0x08,0x61,0x48,0x80, // 'x'
    0x8A,0x45,0x14,0x60,0x82,0x10,0x40, // 'y'
    0xF1,0x08,0x88,0x43,0xC0, // 'z'
    0x64,0x88,0x88,0x88,0x84,0x60, // '{'
    0xAA,0xAA,0xA8, // '|'
    0xC4,0x44,0x42,0x44,0x44,0xC0 // '}'
};
const GFXglyph Roboto_Condensed_Light_12Glyphs[] PROGMEM = {
// bitmapOffset, width, height, xAdvance, xOffset, yOffset
      {     0,   2,   1,   4,    0,   -1 }, // ' '
      {     1,   2,   9,   4,    1,   -9 }, // '!'
      {     4,   3,   3,   4,    1,   -9 }, // '"'
      {     6,   6,   9,   7,    1,   -9 }, // '#'
      {    13,   5,  11,   7,    1,  -10 }, // '$'
      {    20,   7,   9,   9,    1,   -9 }, // '%'
      {    28,   7,   9,   7,    0,   -9 }, // '&'
      {    36,   2,   3,   3,    1,   -9 }, // '''
      {    37,   3,  13,   5,    1,  -10 }, // '('
      {    42,   4,  13,   5,    0,  -10 }, // ')'
      {    49,   6,   5,   6,    0,   -9 }, // '*'
      {    53,   6,   7,   7,    0,   -7 }, // '+'
      {    59,   3,   3,   3,    0,   -1 }, // ','
      {    61,   4,   1,   4,    0,   -4 }, // '-'
      {    62,   2,   1,   4,    1,   -1 }, // '.'
      {    63,   5,  10,   5,    0,   -9 }, // '/'
      {    70,   5,   9,   7,    1,   -9 }, // '0'
      {    76,   4,   9,   7,    1,   -9 }, // '1'
      {    81,   5,   9,   7,    1,   -9 }, // '2'
      {    87,   5,   9,   7,    1,   -9 }, // '3'
      {    93,   7,   9,   7,    0,   -9 }, // '4'
      {   101,   5,   9,   7,    1,   -9 }, // '5'
      {   107,   5,   9,   7,    1,   -9 }, // '6'
      {   113,   6,   9,   7,    0,   -9 }, // '7'
      {   120,   5,   9,   7,    1,   -9 }, // '8'
      {   126,   5,   9,   7,    1,   -9 }, // '9'
      {   132,   2,   6,   3,    1,   -6 }, // ':'
      {   134,   3,   8,   3,    0,   -6 }, // ';'
      {   137,   5,   6,   6,    0,   -7 }, // '<'
      {   141,   5,   3,   7,    1,   -5 }, // '='
      {   143,   5,   6,   6,    1,   -7 }, // '>'
      {   147,   4,   9,   6,    1,   -9 }, // '?'
      {   152,  10,  11,  10,    0,   -8 }, // '@'
      {   166,   8,   9,   8,    0,   -9 }, // 'A'
      {   175,   6,   9,   7,    1,   -9 }, // 'B'
      {   182,   6,   9,   8,    1,   -9 }, // 'C'
      {   189,   6,   9,   8,    1,   -9 }, // 'D'
      {   196,   5,   9,   7,    1,   -9 }, // 'E'
      {   202,   5,   9,   7,    1,   -9 }, // 'F'
      {   208,   6,   9,   8,    1,   -9 }, // 'G'
      {   215,   6,   9,   8,    1,   -9 }, // 'H'
      {   222,   2,   9,   4,    1,   -9 }, // 'I'
      {   225,   5,   9,   7,    1,   -9 }, // 'J'
      {   231,   7,   9,   8,    1,   -9 }, // 'K'
      {   239,   5,   9,   7,    1,   -9 }, // 'L'
      {   245,   8,   9,  10,    1,   -9 }, // 'M'
      {   254,   6,   9,   8,    1,   -9 }, // 'N'
      {   261,   6,   9,   8,    1,   -9 }, // 'O'
      {   268,   6,   9,   7,    1,   -9 }, // 'P'
      {   275,   6,  10,   8,    1,   -9 }, // 'Q'
      {   283,   6,   9,   8,    1,   -9 }, // 'R'
      {   290,   7,   9,   7,    0,   -9 }, // 'S'
      {   298,   7,   9,   7,    0,   -9 }, // 'T'
      {   306,   6,   9,   8,    1,   -9 }, // 'U'
      {   313,   7,   9,   7,    0,   -9 }, // 'V'
      {   321,  10,   9,  10,    0,   -9 }, // 'W'
      {   333,   7,   9,   7,    0,   -9 }, // 'X'
      {   341,   7,   9,   7,    0,   -9 }, // 'Y'
      {   349,   6,   9,   7,    1,   -9 }, // 'Z'
      {   356,   3,  12,   4,    1,  -10 }, // '['
      {   361,   5,  10,   5,    0,   -9 }, // '\'
      {   368,   3,  12,   4,    0,  -10 }, // ']'
      {   373,   5,   4,   5,    0,   -9 }, // '^'
      {   376,   6,   1,   6,    0,    0 }, // '_'
      {   377,   3,   2,   4,    1,   -9 }, // '`'
      {   378,   5,   7,   7,    1,   -7 }, // 'a'
      {   383,   5,   9,   7,    1,   -9 }, // 'b'
      {   389,   6,   7,   6,    0,   -7 }, // 'c'
      {   395,   5,   9,   7,    1,   -9 }, // 'd'
      {   401,   6,   7,   6,    0,   -7 }, // 'e'
      {   407,   4,   9,   5,    1,   -9 }, // 'f'
      {   412,   5,   9,   7,    1,   -7 }, // 'g'
      {   418,   5,   9,   7,    1,   -9 }, // 'h'
      {   424,   2,   9,   4,    1,   -9 }, // 'i'
      {   427,   3,  11,   4,    0,   -9 }, // 'j'
      {   432,   5,   9,   6,    1,   -9 }, // 'k'
      {   438,   2,   9,   4,    1,   -9 }, // 'l'
      {   441,   8,   7,  10,    1,   -7 }, // 'm'
      {   448,   5,   7,   7,    1,   -7 }, // 'n'
      {   453,   5,   7,   7,    1,   -7 }, // 'o'
      {   458,   5,   9,   7,    1,   -7 }, // 'p'
      {   464,   5,   9,   7,    1,   -7 }, // 'q'
      {   470,   4,   7,   5,    1,   -7 }, // 'r'
      {   474,   6,   7,   6,    0,   -7 }, // 's'
      {   480,   4,   9,   4,    0,   -9 }, // 't'
      {   485,   5,   7,   7,    1,   -7 }, // 'u'
      {   490,   6,   7,   6,    0,   -7 }, // 'v'
      {   496,   8,   7,   9,    0,   -7 }, // 'w'
      {   503,   6,   7,   6,    0,   -7 }, // 'x'
      {   509,   6,   9,   6,    0,   -7 }, // 'y'
      {   516,   5,   7,   6,    1,   -7 }, // 'z'
      {   521,   4,  11,   5,    1,   -9 }, // '{'
      {   527,   2,  11,   4,    1,   -9 }, // '|'
      {   530,   4,  11,   5,    0,   -9 } // '}'
};
const GFXfont Roboto_Condensed_Light_12 PROGMEM = {
(uint8_t  *)Roboto_Condensed_Light_12Bitmaps,(GFXglyph *)Roboto_Condensed_Light_12Glyphs,0x20, 0x7E,
15};
```

3.16 roboto9.h

```
// Created by https://oleddisplay.squix.ch/ Consider a donation
// In case of problems make sure that you are using the font file with the correct version!
const uint8_t Roboto_Condensed_Light_9Bitmaps[] PROGMEM = {

    // Bitmap Data:
    0x00, // ' '
    0xAA,0x88, // '!'
    0xA0, // '"'
    0x51,0x47,0x98,0xF1,0x86,0x00, // '#'
    0x64,0xA4,0xC1,0x49,0x88, // '$'
    0xC7,0x38,0x87,0x58,0xC0, // '%'
    0xC6,0x31,0x0E,0x53,0xC0, // '&'
    0xA0, // '''
    0xD2,0x49,0x24,0xC0, // '('
    0x89,0x24,0x92,0x80, // ')'
    0x47,0xA8, // '*'
    0x42,0x3C,0x84,0x00, // '+'
    0xA0, // ','
    0xC0, // '-'
    0x80, // '.'
    0x24,0x44,0x48,0x88, // '/'
    0x64,0xA5,0x29,0x49,0x80, // '0'
    0x59,0x24,0x90, // '1'
    0x64,0x88,0x44,0x43,0xC0, // '2'
    0x64,0x88,0xC1,0x49,0x80, // '3'
    0x21,0x18,0xCA,0x78,0x80, // '4'
    0x74,0x19,0x21,0x49,0x80, // '5'
    0x64,0x39,0x29,0x49,0x80, // '6'
    0xF0,0x88,0x42,0x21,0x00, // '7'
    0x64,0xA8,0xC9,0x49,0x80, // '8'
    0x65,0x25,0x27,0x11,0x80, // '9'
    0x80,0x80, // ':'
    0x80,0xA0, // ';'
    0x24,0xC2, // '<'
    0xE0,0xE0, // '='
    0x84,0x68, // '>'
    0xD9,0x28,0x20, // '?'
    0x38,0x44,0x9C,0xAA,0xAC,0xBC,0x40,0x38, // '@'
    0x21,0x86,0x14,0x51,0xC8,0x80, // 'A'
    0xEA,0xAE,0xAA,0xE0, // 'B'
    0xE5,0x25,0x08,0x5B,0x80, // 'C'
    0xEA,0xAA,0xAA,0xE0, // 'D'
    0xE8,0x8E,0x88,0xE0, // 'E'
    0xE8,0x8E,0x88,0x80, // 'F'
    0x74,0xA1,0x09,0x49,0xC0, // 'G'
    0x94,0xA5,0xE9,0x4A,0x40, // 'H'
    0xAA,0xA8, // 'I'
    0x22,0x22,0x2A,0x60, // 'J'
    0xB5,0x31,0x8C,0x52,0xC0, // 'K'
    0x88,0x88,0x88,0xE0, // 'L'
    0x8A,0x68,0xB6,0xEB,0xAA,0x80, // 'M'
    0x94,0xB5,0xAB,0x5A,0x40, // 'N'
    0x72,0x48,0xA2,0x8A,0x47,0x00, // 'O'
    0xE5,0x25,0x4E,0x42,0x00, // 'P'
    0x72,0x48,0xA2,0x8A,0x47,0x06, // 'Q'
    0xE5,0x29,0x4E,0x52,0xC0, // 'R'
    0xEA,0x84,0x2A,0xE0, // 'S'
    0xF1,0x08,0x42,0x10,0x80, // 'T'
    0xAA,0xAA,0xAA,0xE0, // 'U'
    0x8A,0x49,0x14,0x61,0x82,0x00, // 'V'
    0x92,0xB4,0xB4,0xB4,0x6C,0x4C,0x48, // 'W'
    0x91,0x46,0x08,0x61,0x49,0x80, // 'X'
    0x9A,0x46,0x08,0x20,0x82,0x00, // 'Y'
    0xF0,0x88,0x44,0x43,0xC0, // 'Z'
    0xAA,0xAA, // '['
    0x88,0x84,0x44,0x22, // '\'
    0xAA,0xAA, // ']'
    0x4C,0xA0, // '^'
    0xE0, // '_'
    0x88, // '`'
    0x65,0x19,0x47,0x00, // 'a'
    0x84,0x39,0x29,0x4B,0x80, // 'b'
    0x64,0xA1,0x26,0x00, // 'c'
    0x10,0x9D,0x29,0x49,0xC0, // 'd'
    0x64,0xBD,0x07,0x00, // 'e'
    0x64,0xE4,0x44,0x40, // 'f'
    0x74,0xA5,0x27,0x49,0x80, // 'g'
    0x84,0x39,0x49,0x4A,0x40, // 'h'
    0x8A,0xA8, // 'i'
    0x8A,0xAA,0x80, // 'j'
    0x88,0xE8,0x8C,0xE0, // 'k'
    0xAA,0xA8, // 'l'
    0xDA,0xAA,0xAA,0xA8, // 'm'
    0xE5,0x25,0x29,0x00, // 'n'
    0x64,0xA5,0x26,0x00, // 'o'
    0xE4,0xA5,0x2E,0x42,0x00, // 'p'
    0x74,0xA5,0x27,0x08,0x40, // 'q'
    0xD2,0x48, // 'r'
    0x64,0x18,0x4F,0x00, // 's'
    0x59,0x24,0x80, // 't'
    0x94,0xA5,0x27,0x00, // 'u'
    0x95,0x18,0xC4,0x00, // 'v'
    0xAD,0x72,0xE2,0x45,0x00, // 'w'
    0xB3,0x10,0xCB,0x00, // 'x'
    0x95,0x18,0xC4,0x21,0x00, // 'y'
    0xE1,0x11,0x0F,0x00, // 'z'
    0x64,0x44,0x84,0x44,0x60, // '{'
    0xAA,0xAA, // '|'
    0x84,0x44,0x24,0x44,0x80 // '}'
};
const GFXglyph Roboto_Condensed_Light_9Glyphs[] PROGMEM = {
// bitmapOffset, width, height, xAdvance, xOffset, yOffset
      {     0,   2,   1,   3,    0,   -1 }, // ' '
      {     1,   2,   7,   3,    1,   -7 }, // '!'
      {     3,   2,   2,   4,    1,   -7 }, // '"'
      {     4,   6,   7,   6,    0,   -7 }, // '#'
      {    10,   5,   8,   5,    0,   -7 }, // '$'
      {    15,   5,   7,   7,    1,   -7 }, // '%'
      {    20,   5,   7,   6,    1,   -7 }, // '&'
      {    25,   2,   2,   3,    0,   -7 }, // '''
      {    26,   3,   9,   4,    1,   -7 }, // '('
      {    30,   3,   9,   4,    0,   -7 }, // ')'
      {    34,   5,   3,   5,    0,   -7 }, // '*'
      {    36,   5,   5,   5,    0,   -5 }, // '+'
      {    40,   2,   2,   3,    0,   -1 }, // ','
      {    41,   3,   1,   3,    0,   -3 }, // '-'
      {    42,   2,   1,   3,    0,   -1 }, // '.'
      {    43,   4,   8,   4,    0,   -7 }, // '/'
      {    47,   5,   7,   5,    0,   -7 }, // '0'
      {    52,   3,   7,   5,    1,   -7 }, // '1'
      {    55,   5,   7,   5,    0,   -7 }, // '2'
      {    60,   5,   7,   5,    0,   -7 }, // '3'
      {    65,   5,   7,   5,    0,   -7 }, // '4'
      {    70,   5,   7,   5,    0,   -7 }, // '5'
```

```
        {    75,   5,   7,   5,    0,   -7 }, // ’6’
        {    80,   5,   7,   5,    0,   -7 }, // ’7’
        {    85,   5,   7,   5,    0,   -7 }, // ’8’
        {    90,   5,   7,   5,    0,   -7 }, // ’9’
        {    95,   2,   5,   3,    0,   -5 }, // ’:’
        {    97,   2,   6,   3,    0,   -5 }, // ’;’
        {    99,   4,   4,   5,    0,   -5 }, // ’<’
        {   101,   4,   3,   5,    1,   -4 }, // ’=’
        {   103,   4,   4,   5,    1,   -5 }, // ’>’
        {   105,   3,   7,   5,    1,   -7 }, // ’?’
        {   108,   8,   8,   8,    0,   -6 }, // ’@’
        {   116,   6,   7,   6,    0,   -7 }, // ’A’
        {   122,   4,   7,   6,    1,   -7 }, // ’B’
        {   126,   5,   7,   6,    1,   -7 }, // ’C’
        {   131,   4,   7,   6,    1,   -7 }, // ’D’
        {   135,   4,   7,   5,    1,   -7 }, // ’E’
        {   139,   4,   7,   5,    1,   -7 }, // ’F’
        {   143,   5,   7,   6,    0,   -7 }, // ’G’
        {   148,   5,   7,   7,    1,   -7 }, // ’H’
        {   153,   2,   7,   3,    1,   -7 }, // ’I’
        {   155,   4,   7,   5,    0,   -7 }, // ’J’
        {   159,   5,   7,   6,    1,   -7 }, // ’K’
        {   164,   4,   7,   5,    1,   -7 }, // ’L’
        {   168,   6,   7,   8,    1,   -7 }, // ’M’
        {   174,   5,   7,   7,    1,   -7 }, // ’N’
        {   179,   6,   7,   6,    0,   -7 }, // ’O’
        {   185,   5,   7,   6,    1,   -7 }, // ’P’
        {   190,   6,   8,   6,    0,   -7 }, // ’Q’
        {   196,   5,   7,   6,    1,   -7 }, // ’R’
        {   201,   4,   7,   6,    1,   -7 }, // ’S’
        {   205,   5,   7,   6,    0,   -7 }, // ’T’
        {   210,   4,   7,   6,    1,   -7 }, // ’U’
        {   214,   6,   7,   6,    0,   -7 }, // ’V’
        {   220,   8,   7,   8,    0,   -7 }, // ’W’
        {   227,   6,   7,   6,    0,   -7 }, // ’X’
        {   233,   6,   7,   6,    0,   -7 }, // ’Y’
        {   239,   5,   7,   6,    0,   -7 }, // ’Z’
        {   244,   2,   8,   3,    1,   -7 }, // ’[’
        {   246,   4,   8,   4,    0,   -7 }, // ’\’
        {   250,   2,   8,   3,    0,   -7 }, // ’]’
        {   252,   4,   3,   4,    0,   -7 }, // ’^’
        {   254,   4,   1,   4,    0,    0 }, // ’_’
        {   255,   3,   2,   4,    0,   -8 }, // ’‘’
        {   256,   5,   5,   5,    0,   -5 }, // ’a’
        {   260,   5,   7,   5,    0,   -7 }, // ’b’
        {   265,   5,   5,   5,    0,   -5 }, // ’c’
        {   269,   5,   7,   5,    0,   -7 }, // ’d’
        {   274,   5,   5,   5,    0,   -5 }, // ’e’
        {   278,   4,   7,   4,    0,   -7 }, // ’f’
        {   282,   5,   7,   5,    0,   -5 }, // ’g’
        {   287,   5,   7,   5,    0,   -7 }, // ’h’
        {   292,   2,   7,   3,    1,   -7 }, // ’i’
        {   294,   2,   9,   3,    0,   -7 }, // ’j’
        {   297,   4,   7,   5,    1,   -7 }, // ’k’
        {   301,   2,   7,   3,    0,   -7 }, // ’l’
        {   303,   6,   5,   8,    1,   -5 }, // ’m’
        {   307,   5,   5,   5,    0,   -5 }, // ’n’
        {   311,   5,   5,   5,    0,   -5 }, // ’o’
        {   315,   5,   7,   5,    0,   -5 }, // ’p’
        {   320,   5,   7,   5,    0,   -5 }, // ’q’
        {   325,   3,   5,   4,    1,   -5 }, // ’r’
        {   327,   5,   5,   5,    0,   -5 }, // ’s’
        {   331,   3,   6,   4,    0,   -6 }, // ’t’
        {   334,   5,   5,   5,    0,   -5 }, // ’u’
        {   338,   5,   5,   5,    0,   -5 }, // ’v’
        {   342,   7,   5,   7,    0,   -5 }, // ’w’
        {   347,   5,   5,   5,    0,   -5 }, // ’x’
        {   351,   5,   7,   5,    0,   -5 }, // ’y’
        {   356,   5,   5,   5,    0,   -5 }, // ’z’
        {   360,   4,   9,   4,    0,   -7 }, // ’{’
        {   365,   2,   8,   3,    1,   -7 }, // ’|’
        {   367,   4,   9,   4,    0,   -7 } // ’}’
};
const GFXfont Roboto_Condensed_Light_9 PROGMEM = {
(uint8_t  *)Roboto_Condensed_Light_9Bitmaps,(GFXglyph *)Roboto_Condensed_Light_9Glyphs,0x20, 0x7E, 12};
```

3.17 SansSerif12.h

```
// Created by https://oleddisplay.squix.ch/ Consider a donation
// In case of problems make sure that you are using the font file with the correct version!
const uint8_t SansSerif_plain_12Bitmaps[] PROGMEM = {

```

```
// Bitmap Data:
0x00, // ' '
0xAA,0xA2,0x80, // '!'
0xAA,0xA0, // '"'
0x12,0x0A,0x1F,0xC4,0x82,0x47,0xF0,0xA0,0x90, // '#'
0x21,0xCA,0xA8,0xE0,0xE2,0xAA,0x70,0x82,0x00, // '$'
0x61,0x12,0x42,0x48,0x4A,0x06,0xD8,0x14,0x84,0x90,0x92,0x21,0x80, // '%'
0x30,0x24,0x10,0x0C,0x05,0x14,0x4A,0x19,0x88,0x7B,0x00, // '&'
0xA8, // '''
0x64,0x48,0x88,0x88,0x44,0x60, // '('
0xC4,0x42,0x22,0x22,0x44,0xC0, // ')'
0x22,0xA7,0x1C,0xA8,0x80, // '*'
0x10,0x10,0x10,0xFE,0x10,0x10,0x10, // '+'
0xA8, // ','
0xE0, // '-'
0xA0, // '.'
0x10,0x88,0x42,0x21,0x08,0x84,0x00, // '/'
0x78,0x92,0x14,0x28,0x50,0xA1,0x24,0x78, // '0'
0xE0,0x82,0x08,0x20,0x82,0x08,0xF8, // '1'
0x79,0x18,0x10,0x20,0x82,0x08,0x20,0xFC, // '2'
0x79,0x08,0x10,0x23,0x80,0x81,0x42,0x78, // '3'
0x18,0x30,0xA2,0x44,0x91,0x3F,0x04,0x08, // '4'
0xF9,0x02,0x07,0xC0,0xC0,0x81,0x46,0x78, // '5'
0x38,0x8A,0x05,0xCC,0xD0,0xA1,0x26,0x78, // '6'
0xFC,0x08,0x20,0x41,0x02,0x08,0x10,0x40, // '7'
0x79,0x0A,0x14,0x27,0x90,0xA1,0x42,0x78, // '8'
0x79,0x92,0x14,0x2C,0xCE,0x81,0x44,0x70, // '9'
0xA0,0xA0, // ':'
0xA0,0xA8, // ';'
0x03,0x0F,0x38,0x1C,0x01,0xE0,0x18, // '<'
0xFF,0x00,0x3F,0xC0, // '='
0xC0,0x3C,0x01,0xC0,0xE7,0x86,0x00, // '>'
0x72,0x20,0x84,0x20,0x80,0x08,0x20, // '?'
0x1F,0x02,0x0C,0x40,0x48,0xF2,0x91,0x29,0x12,0x91,0x48,0xF8,0x40,0x02,0x08,0x1F,0x00, // '@'
0x18,0x0C,0x09,0x04,0x82,0x42,0x11,0xF8,0x84,0x81,0x00, // 'A'
0xF9,0x0A,0x14,0x2F,0x90,0xA1,0x42,0xF8, // 'B'
0x38,0x8A,0x04,0x08,0x10,0x20,0x22,0x38, // 'C'
0xF8,0x84,0x82,0x82,0x82,0x82,0x82,0x84,0xF8, // 'D'
0xFD,0x02,0x04,0x0F,0xD0,0x20,0x40,0xFC, // 'E'
0xFA,0x08,0x20,0xFA,0x08,0x20,0x80, // 'F'
0x3C,0x42,0x80,0x80,0x8E,0x82,0x82,0x42,0x3C, // 'G'
0x82,0x82,0x82,0x82,0xFE,0x82,0x82,0x82,0x82, // 'H'
0xAA,0xAA,0x80, // 'I'
0x22,0x22,0x22,0x22,0x22,0xC0, // 'J'
0x84,0x88,0x90,0xA0,0xC0,0xA0,0x90,0x88,0x84, // 'K'
0x82,0x08,0x20,0x82,0x08,0x20,0xF8, // 'L'
0x81,0x61,0xB0,0xD4,0xAA,0x54,0xCA,0x65,0x02,0x81,0x00, // 'M'
0xC2,0xC2,0xA2,0xA2,0x92,0x8A,0x8A,0x86,0x86, // 'N'
0x38,0x44,0x82,0x82,0x82,0x82,0x82,0x44,0x38, // 'O'
0xF9,0x0A,0x14,0x2F,0x90,0x20,0x40,0x80, // 'P'
0x38,0x44,0x82,0x82,0x82,0x82,0x82,0x44,0x38,0x08,0x04, // 'Q'
0xF8,0x84,0x84,0x84,0xF8,0x88,0x84,0x84,0x82, // 'R'
0x79,0x0A,0x04,0x07,0x80,0x81,0x42,0x78, // 'S'
0xFE,0x10,0x10,0x10,0x10,0x10,0x10,0x10,0x10, // 'T'
0x82,0x82,0x82,0x82,0x82,0x82,0x82,0xC6,0x7C, // 'U'
0x40,0x88,0x10,0x84,0x10,0x82,0x10,0x24,0x04,0x80,0x60,0x0C,0x00, // 'V'
0x84,0x24,0x44,0x44,0x44,0xA4,0x2A,0x82,0xA8,0x2A,0x81,0x10,0x11,0x00, // 'W'
0xC6,0x44,0x28,0x28,0x10,0x28,0x28,0x44,0x82, // 'X'
0x82,0x44,0x44,0x28,0x28,0x10,0x10,0x10,0x10, // 'Y'
0xFE,0x02,0x04,0x08,0x10,0x20,0x40,0x80,0xFE, // 'Z'
0xD2,0x49,0x24,0x93,0x00, // '['
0x84,0x10,0x84,0x10,0x84,0x10,0x80, // '\'
0xC9,0x24,0x92,0x4B,0x00, // ']'
0x18,0x12,0x10,0x80, // '^'
0xFC, // '_'
0x42, // '`'
0x79,0x08,0x13,0xE8,0x51,0x9D,0x00, // 'a'
0x81,0x02,0x07,0xCC,0xD0,0xA1,0x42,0xCD,0xF0, // 'b'
0x73,0x28,0x20,0x83,0x27,0x00, // 'c'
0x04,0x08,0x13,0xEC,0xD0,0xA1,0x42,0xCC,0xF8, // 'd'
0x79,0x9A,0x17,0xE8,0x18,0x9E,0x00, // 'e'
0x32,0x11,0xE4,0x21,0x08,0x42,0x00, // 'f'
0x7D,0x9A,0x14,0x28,0x59,0x9F,0x02,0x4C,0x70, // 'g'
0x81,0x02,0x05,0xCC,0x50,0xA1,0x42,0x85,0x08, // 'h'
0x8A,0xAA,0x80, // 'i'
0x41,0x24,0x92,0x49,0x60, // 'j'
0x81,0x02,0x04,0x49,0x14,0x30,0x50,0x91,0x10, // 'k'
0xAA,0xAA,0xA0, // 'l'
0xF7,0x22,0x28,0x8A,0x22,0x88,0xA2,0x28,0x88, // 'm'
0xB9,0x8A,0x14,0x28,0x50,0xA1,0x00, // 'n'
0x79,0x9A,0x14,0x28,0x59,0x9E,0x00, // 'o'
0xF9,0x9A,0x14,0x28,0x59,0xBE,0x40,0x81,0x00, // 'p'
0x7D,0x9A,0x14,0x28,0x59,0x9F,0x02,0x04,0x08, // 'q'
0xB6,0x21,0x08,0x42,0x00, // 'r'
0x72,0x28,0x1C,0x0A,0x27,0x00, // 's'
0x42,0x3C,0x84,0x21,0x08,0x70, // 't'
0x85,0x0A,0x14,0x28,0x51,0x9D,0x00, // 'u'
```

```
    0x85,0x09,0x22,0x44,0x86,0x0C,0x00, // 'v'
    0x88,0xA2,0x25,0x51,0x54,0x55,0x08,0x82,0x20, // 'w'
    0x84,0x91,0x21,0x84,0x89,0x21,0x00, // 'x'
    0x85,0x09,0x22,0x42,0x86,0x04,0x08,0x21,0x80, // 'y'
    0xF8,0x21,0x08,0x42,0x0F,0x80, // 'z'
    0x38,0x82,0x08,0x23,0x02,0x08,0x20,0x83,0x80, // '{'
    0xAA,0xAA,0xAA, // '|'
    0xE0,0x82,0x08,0x20,0x62,0x08,0x20,0x8E,0x00 // '}'
};
const GFXglyph SansSerif_plain_12Glyphs[] PROGMEM = {
// bitmapOffset, width, height, xAdvance, xOffset, yOffset
      {     0,   2,   1,   5,    0,   -1 }, // ' '
      {     1,   2,   9,   6,    2,   -9 }, // '!'
      {     4,   4,   3,   6,    1,   -9 }, // '"'
      {     6,   9,   8,  11,    1,   -8 }, // '#'
      {    15,   6,  11,   9,    2,   -9 }, // '$'
      {    24,  11,   9,  12,    0,   -9 }, // '%'
      {    37,   9,   9,  11,    1,   -9 }, // '&'
      {    48,   2,   3,   4,    1,   -9 }, // '''
      {    49,   4,  11,   6,    1,  -10 }, // '('
      {    55,   4,  11,   6,    1,  -10 }, // ')'
      {    61,   6,   6,   7,    1,   -9 }, // '*'
      {    66,   8,   7,  11,    1,   -7 }, // '+'
      {    73,   2,   3,   5,    1,   -2 }, // ','
      {    74,   4,   1,   5,    1,   -4 }, // '-'
      {    75,   2,   2,   5,    1,   -2 }, // '.'
      {    76,   5,  10,   5,    0,   -9 }, // '/'
      {    83,   7,   9,   9,    1,   -9 }, // '0'
      {    91,   6,   9,   9,    1,   -9 }, // '1'
      {    98,   7,   9,   9,    1,   -9 }, // '2'
      {   106,   7,   9,   9,    1,   -9 }, // '3'
      {   114,   7,   9,   9,    1,   -9 }, // '4'
      {   122,   7,   9,   9,    1,   -9 }, // '5'
      {   130,   7,   9,   9,    1,   -9 }, // '6'
      {   138,   7,   9,   9,    1,   -9 }, // '7'
      {   146,   7,   9,   9,    1,   -9 }, // '8'
      {   154,   7,   9,   9,    1,   -9 }, // '9'
      {   162,   2,   6,   5,    1,   -6 }, // ':'
      {   164,   2,   7,   5,    1,   -6 }, // ';'
      {   166,   9,   6,  11,    1,   -7 }, // '<'
      {   173,   9,   3,  11,    1,   -5 }, // '='
      {   177,   9,   6,  11,    1,   -7 }, // '>'
      {   184,   6,   9,   7,    0,   -9 }, // '?'
      {   191,  12,  11,  14,    1,   -9 }, // '@'
      {   208,   9,   9,   9,    0,   -9 }, // 'A'
      {   219,   7,   9,   9,    1,   -9 }, // 'B'
      {   227,   7,   9,   9,    1,   -9 }, // 'C'
      {   235,   8,   9,  10,    1,   -9 }, // 'D'
      {   244,   7,   9,   9,    1,   -9 }, // 'E'
      {   252,   6,   9,   8,    1,   -9 }, // 'F'
      {   259,   8,   9,  10,    1,   -9 }, // 'G'
      {   268,   8,   9,  10,    1,   -9 }, // 'H'
      {   277,   2,   9,   4,    1,   -9 }, // 'I'
      {   280,   4,  11,   4,   -1,   -9 }, // 'J'
      {   286,   8,   9,   8,    1,   -9 }, // 'K'
      {   295,   6,   9,   7,    1,   -9 }, // 'L'
      {   302,   9,   9,  11,    1,   -9 }, // 'M'
      {   313,   8,   9,  10,    1,   -9 }, // 'N'
      {   322,   8,   9,  10,    1,   -9 }, // 'O'
      {   331,   7,   9,   9,    1,   -9 }, // 'P'
      {   339,   8,  11,  10,    1,   -9 }, // 'Q'
      {   350,   8,   9,   9,    1,   -9 }, // 'R'
      {   359,   7,   9,   9,    1,   -9 }, // 'S'
      {   367,   8,   9,   8,    0,   -9 }, // 'T'
      {   376,   8,   9,  10,    1,   -9 }, // 'U'
      {   385,  11,   9,   9,   -1,   -9 }, // 'V'
      {   398,  12,   9,  12,    0,   -9 }, // 'W'
      {   412,   8,   9,   8,    0,   -9 }, // 'X'
      {   421,   8,   9,   8,    0,   -9 }, // 'Y'
      {   430,   8,   9,  10,    1,   -9 }, // 'Z'
      {   439,   3,  11,   6,    2,   -9 }, // '['
      {   444,   5,  10,   5,    0,   -9 }, // '\'
      {   451,   3,  11,   6,    1,   -9 }, // ']'
      {   456,   9,   3,  11,    1,   -9 }, // '^'
      {   460,   7,   1,   7,    0,    2 }, // '_'
      {   461,   4,   2,   7,    1,  -10 }, // '`'
      {   462,   7,   7,   9,    1,   -7 }, // 'a'
      {   469,   7,  10,   9,    1,  -10 }, // 'b'
      {   478,   6,   7,   8,    1,   -7 }, // 'c'
      {   484,   7,  10,   9,    1,  -10 }, // 'd'
      {   493,   7,   7,   9,    1,   -7 }, // 'e'
      {   500,   5,  10,   5,    0,  -10 }, // 'f'
      {   507,   7,  10,   9,    1,   -7 }, // 'g'
      {   516,   7,  10,   9,    1,  -10 }, // 'h'
      {   525,   2,   9,   4,    1,   -9 }, // 'i'
      {   528,   3,  12,   4,    0,   -9 }, // 'j'
      {   533,   7,  10,   8,    1,  -10 }, // 'k'
```

```
        {   542,   2,  10,   4,    1,  -10 }, // 'l'
        {   545,  10,   7,  12,    1,   -7 }, // 'm'
        {   554,   7,   7,   9,    1,   -7 }, // 'n'
        {   561,   7,   7,   9,    1,   -7 }, // 'o'
        {   568,   7,  10,   9,    1,   -7 }, // 'p'
        {   577,   7,  10,   9,    1,   -7 }, // 'q'
        {   586,   5,   7,   6,    1,   -7 }, // 'r'
        {   591,   6,   7,   8,    1,   -7 }, // 's'
        {   597,   5,   9,   6,    0,   -9 }, // 't'
        {   603,   7,   7,   9,    1,   -7 }, // 'u'
        {   610,   7,   7,   7,    0,   -7 }, // 'v'
        {   617,  10,   7,  10,    0,   -7 }, // 'w'
        {   626,   7,   7,   7,    0,   -7 }, // 'x'
        {   633,   7,  10,   7,    0,   -7 }, // 'y'
        {   642,   6,   7,   6,    0,   -7 }, // 'z'
        {   648,   6,  11,   9,    2,   -9 }, // '{'
        {   657,   2,  12,   5,    2,   -9 }, // '|'
        {   660,   6,  11,   9,    1,   -9 } // '}'
};
const GFXfont SansSerif_plain_12 PROGMEM = {
(uint8_t  *)SansSerif_plain_12Bitmaps,(GFXglyph *)SansSerif_plain_12Glyphs,0x20, 0x7E, 15};
```

3.18 SansSerif9.h

```
// Created by https://oleddisplay.squix.ch/ Consider a donation
// In case of problems make sure that you are using the font file with the correct version!
const uint8_t SansSerif_plain_9Bitmaps[] PROGMEM = {

    // Bitmap Data:
    0x00, // ' '
    0xAA,0x88, // '!'
    0xAA, // '"'
    0x28,0x53,0xF1,0x4F,0xCA,0x14,0x00, // '#'
    0x21,0xEA,0x38,0x38,0xAF,0x08, // '$'
    0xE4,0x52,0x2A,0x1F,0xE1,0x51,0x28,0x9C, // '%'
    0x30,0x91,0x03,0x29,0x53,0x1B,0x00, // '&'
    0xA0, // '''
    0x52,0x49,0x22, // '('
    0x91,0x24,0xA4, // ')'
    0xA9,0xC7,0x2A, // '*'
    0x20,0x8F,0x88,0x20, // '+'
    0xA0, // ','
    0xC0, // '-'
    0x80, // '.'
    0x22,0x44,0x48,0x80, // '/'
    0x64,0xA5,0x29,0x49,0x80, // '0'
    0xC4,0x44,0x44,0xE0, // '1'
    0x64,0x84,0x44,0x43,0xC0, // '2'
    0x64,0x84,0xC1,0x0B,0x80, // '3'
    0x10,0xC5,0x14,0x93,0xE1,0x00, // '4'
    0xF4,0x21,0xC1,0x0B,0x80, // '5'
    0x76,0x21,0xC9,0x49,0x80, // '6'
    0xF0,0x88,0x42,0x21,0x00, // '7'
    0x64,0xA4,0xC9,0x49,0x80, // '8'
    0x64,0xA5,0xE1,0x1B,0x80, // '9'
    0x80,0x80, // ':'
    0x80,0xA0, // ';'
    0x04,0x73,0x01,0xC0,0x40, // '<'
    0xFC,0x03,0xF0, // '='
    0x80,0xE0,0x33,0x88,0x00, // '>'
    0xF0,0x88,0x84,0x01,0x00, // '?'
    0x3C,0x21,0x27,0x54,0xAA,0x54,0xF1,0x10,0x70, // '@'
    0x30,0x61,0x22,0x47,0x90,0xA1,0x00, // 'A'
    0xF2,0x28,0xBC,0x8A,0x2F,0x00, // 'B'
    0x73,0x28,0x20,0x83,0x07,0x80, // 'C'
    0xF2,0x68,0xA2,0x8A,0x6F,0x00, // 'D'
    0xF4,0x21,0xE8,0x43,0xC0, // 'E'
    0xF4,0x21,0xE8,0x42,0x00, // 'F'
    0x73,0x28,0x26,0x8B,0x27,0x00, // 'G'
    0x8A,0x28,0xBE,0x8A,0x28,0x80, // 'H'
    0xAA,0xA8, // 'I'
    0x49,0x24,0x92,0x80, // 'J'
    0x8A,0x4A,0x30,0xA2,0x48,0x80, // 'K'
    0x84,0x21,0x08,0x43,0xC0, // 'L'
    0x85,0x9B,0x35,0xAB,0x50,0xA1,0x00, // 'M'
    0x8B,0x2C,0xAA,0x9A,0x68,0x80, // 'N'
    0x73,0x68,0xA2,0x8B,0x67,0x00, // 'O'
    0xE4,0xA5,0xC8,0x42,0x00, // 'P'
    0x73,0x68,0xA2,0x8B,0x67,0x04, // 'Q'
    0xF2,0x49,0x38,0xA2,0x48,0x80, // 'R'
    0x72,0x28,0x1C,0x0A,0x27,0x00, // 'S'
    0xF8,0x82,0x08,0x20,0x82,0x00, // 'T'
    0x8A,0x28,0xA2,0x8A,0x27,0x00, // 'U'
    0x85,0x09,0x22,0x44,0x86,0x0C,0x00, // 'V'
    0x92,0x92,0x54,0x54,0x6C,0x28,0x28, // 'W'
    0xCC,0x90,0xC1,0x83,0x09,0x23,0x00, // 'X'
    0x89,0x45,0x08,0x20,0x82,0x00, // 'Y'
    0xF8,0x21,0x08,0x42,0x0F,0x80, // 'Z'
    0xD2,0x49,0x26, // '['
    0x88,0x44,0x42,0x20, // '\'
    0xC9,0x24,0x96, // ']'
    0x30,0x90, // '^'
    0xF8, // '_'
    0x88, // '`'
    0x70,0xBD,0x2F,0x00, // 'a'
    0x84,0x21,0xC9,0x4A,0x5C, // 'b'
    0x74,0x21,0x07,0x00, // 'c'
    0x10,0x84,0xE9,0x4A,0x4E, // 'd'
    0x64,0xBD,0x07,0x00, // 'e'
    0x72,0x11,0xC4,0x21,0x08, // 'f'
    0x74,0xA5,0x27,0x09,0x80, // 'g'
    0x84,0x21,0xE9,0x4A,0x52, // 'h'
    0x8A,0xA8, // 'i'
    0x41,0x24,0x92,0xC0, // 'j'
    0x84,0x21,0x2A,0x62,0x92, // 'k'
    0xAA,0xAA, // 'l'
    0xFE,0x92,0x92,0x92,0x92, // 'm'
    0xF4,0xA5,0x29,0x00, // 'n'
    0x64,0xA5,0x26,0x00, // 'o'
    0xE4,0xA5,0x2E,0x42,0x00, // 'p'
    0x74,0xA5,0x27,0x08,0x40, // 'q'
    0xE8,0x88,0x80, // 'r'
    0xE8,0x62,0xE0, // 's'
    0x47,0x90,0x84,0x38, // 't'
    0x94,0xA5,0x2F,0x00, // 'u'
    0x8A,0x25,0x14,0x20, // 'v'
    0x92,0xAA,0xAA,0x44,0x44, // 'w'
    0x89,0x42,0x14,0x88, // 'x'
    0x89,0x11,0x42,0x82,0x04,0x30,0x00, // 'y'
    0xF0,0x88,0x8F,0x00, // 'z'
    0x64,0x48,0x44,0x46, // '{'
    0xAA,0xAA,0x80, // '|'
    0xC4,0x42,0x44,0x4C // '}'
};
const GFXglyph SansSerif_plain_9Glyphs[] PROGMEM = {
// bitmapOffset, width, height, xAdvance, xOffset, yOffset
      {     0,    2,    1,    4,     0,    -1 }, // ' '
      {     1,    2,    7,    4,     1,    -7 }, // '!'
      {     3,    4,    2,    5,     1,    -7 }, // '"'
      {     4,    7,    7,    9,     1,    -7 }, // '#'
      {    11,    6,    8,    7,     0,    -7 }, // '$'
      {    17,    9,    7,   10,     0,    -7 }, // '%'
      {    25,    7,    7,    9,     1,    -7 }, // '&'
      {    32,    2,    2,    3,     1,    -7 }, // '''
      {    33,    3,    8,    5,     1,    -8 }, // '('
      {    36,    3,    8,    5,     1,    -8 }, // ')'
      {    39,    6,    4,    6,     0,    -7 }, // '*'
      {    42,    6,    5,    9,     1,    -5 }, // '+'
      {    46,    2,    2,    4,     1,    -1 }, // ','
      {    47,    3,    1,    4,     1,    -3 }, // '-'
      {    48,    2,    1,    4,     1,    -1 }, // '.'
      {    49,    4,    7,    4,     0,    -7 }, // '/'
      {    53,    5,    7,    7,     1,    -7 }, // '0'
      {    58,    4,    7,    7,     2,    -7 }, // '1'
      {    62,    5,    7,    7,     1,    -7 }, // '2'
      {    67,    5,    7,    7,     1,    -7 }, // '3'
      {    72,    6,    7,    7,     1,    -7 }, // '4'
      {    78,    5,    7,    7,     1,    -7 }, // '5'
      {    83,    5,    7,    7,     1,    -7 }, // '6'
      {    88,    5,    7,    7,     1,    -7 }, // '7'
      {    93,    5,    7,    7,     1,    -7 }, // '8'
      {    98,    5,    7,    7,     1,    -7 }, // '9'
      {   103,    2,    5,    4,     1,    -5 }, // ':'
      {   105,    2,    6,    4,     1,    -5 }, // ';'
      {   107,    7,    5,    9,     1,    -5 }, // '<'
      {   112,    7,    3,    9,     1,    -4 }, // '='
      {   115,    7,    5,    9,     1,    -5 }, // '>'
      {   120,    5,    7,    6,     1,    -7 }, // '?'
      {   125,    9,    8,   11,     1,    -7 }, // '@'
      {   134,    7,    7,    7,     0,    -7 }, // 'A'
      {   141,    6,    7,    8,     1,    -7 }, // 'B'
      {   147,    6,    7,    8,     1,    -7 }, // 'C'
      {   153,    6,    7,    8,     1,    -7 }, // 'D'
      {   159,    5,    7,    7,     1,    -7 }, // 'E'
      {   164,    5,    7,    7,     1,    -7 }, // 'F'
      {   169,    6,    7,    8,     1,    -7 }, // 'G'
      {   175,    6,    7,    8,     1,    -7 }, // 'H'
      {   181,    2,    7,    4,     1,    -7 }, // 'I'
      {   183,    3,    9,    4,     0,    -7 }, // 'J'
      {   187,    6,    7,    7,     1,    -7 }, // 'K'
```

```
         {   193,   5,   7,   6,    1,   -7 }, // ’L’
         {   198,   7,   7,   9,    1,   -7 }, // ’M’
         {   205,   6,   7,   8,    1,   -7 }, // ’N’
         {   211,   6,   7,   8,    1,   -7 }, // ’O’
         {   217,   5,   7,   7,    1,   -7 }, // ’P’
         {   222,   6,   8,   8,    1,   -7 }, // ’Q’
         {   228,   6,   7,   7,    1,   -7 }, // ’R’
         {   234,   6,   7,   8,    1,   -7 }, // ’S’
         {   240,   6,   7,   6,    0,   -7 }, // ’T’
         {   246,   6,   7,   8,    1,   -7 }, // ’U’
         {   252,   7,   7,   7,    0,   -7 }, // ’V’
         {   259,   8,   7,   8,    0,   -7 }, // ’W’
         {   266,   7,   7,   7,    0,   -7 }, // ’X’
         {   273,   6,   7,   6,    0,   -7 }, // ’Y’
         {   279,   6,   7,   6,    0,   -7 }, // ’Z’
         {   285,   3,   8,   5,    1,   -7 }, // ’[’
         {   288,   4,   7,   4,    0,   -7 }, // ’\’
         {   292,   3,   8,   5,    1,   -7 }, // ’]’
         {   295,   7,   2,   9,    1,   -7 }, // ’^’
         {   297,   6,   1,   6,    0,    1 }, // ’_’
         {   298,   3,   2,   6,    1,   -8 }, // ’‘’
         {   299,   5,   5,   7,    1,   -5 }, // ’a’
         {   303,   5,   8,   7,    1,   -8 }, // ’b’
         {   308,   5,   5,   7,    1,   -5 }, // ’c’
         {   312,   5,   8,   7,    1,   -8 }, // ’d’
         {   317,   5,   5,   7,    1,   -5 }, // ’e’
         {   321,   5,   8,   4,    0,   -8 }, // ’f’
         {   326,   5,   7,   7,    1,   -5 }, // ’g’
         {   331,   5,   8,   7,    1,   -8 }, // ’h’
         {   336,   2,   7,   4,    1,   -7 }, // ’i’
         {   338,   3,   9,   4,    0,   -7 }, // ’j’
         {   342,   5,   8,   6,    1,   -8 }, // ’k’
         {   347,   2,   8,   4,    1,   -8 }, // ’l’
         {   349,   8,   5,  10,    1,   -5 }, // ’m’
         {   354,   5,   5,   7,    1,   -5 }, // ’n’
         {   358,   5,   5,   7,    1,   -5 }, // ’o’
         {   362,   5,   7,   7,    1,   -5 }, // ’p’
         {   367,   5,   7,   7,    1,   -5 }, // ’q’
         {   372,   4,   5,   5,    1,   -5 }, // ’r’
         {   375,   4,   5,   6,    1,   -5 }, // ’s’
         {   378,   5,   6,   5,    0,   -6 }, // ’t’
         {   382,   5,   5,   7,    1,   -5 }, // ’u’
         {   386,   6,   5,   6,    0,   -5 }, // ’v’
         {   390,   8,   5,   8,    0,   -5 }, // ’w’
         {   395,   6,   5,   6,    0,   -5 }, // ’x’
         {   399,   7,   7,   6,    0,   -5 }, // ’y’
         {   406,   5,   5,   7,    1,   -5 }, // ’z’
         {   410,   4,   8,   6,    1,   -7 }, // ’{’
         {   414,   2,   9,   4,    1,   -7 }, // ’|’
         {   417,   4,   8,   6,    1,   -7 } // ’}’
};
const GFXfont SansSerif_plain_9 PROGMEM = {
(uint8_t  *)SansSerif_plain_9Bitmaps,(GFXglyph *)SansSerif_plain_9Glyphs,0x20, 0x7E, 12};
```

3.19 serif12.h

```
// Created by https://oleddisplay.squix.ch/ Consider a donation
// In case of problems make sure that you are using the font file with the correct version!
const uint8_t Serif_plain_12Bitmaps[] PROGMEM = {

    // Bitmap Data:
    0x00, // ’ ’
    0xDB,0x6D,0x86,0xC0, // ’!’
    0xAA,0xA0, // ’"’
    0x12,0x0A,0x1F,0xC2,0x82,0x41,0x23,0xF8,0x50,0x48,0x00, // ’#’
    0x20,0x41,0xE5,0x6A,0x4C,0x0E,0x52,0xA4,0xF0,0x81,0x00, // ’$’
    0x61,0x12,0x42,0x48,0x4A,0x06,0xD8,0x14,0x84,0x90,0x92,0x21,0x80, // ’%’
    0x38,0x11,0x04,0x01,0x80,0x93,0xA2,0x48,0x63,0x08,0x7D,0x80, // ’&’
    0xA8, // ’’’
    0x24,0x88,0x88,0x88,0x84,0x20, // ’(’
    0x84,0x42,0x22,0x22,0x44,0x80, // ’)’
    0x22,0xA7,0x3E,0x20, // ’*’
    0x10,0x10,0x10,0xFE,0x10,0x10,0x10, // ’+’
    0x4A,0x00, // ’,’
    0xE0, // ’-’
    0xD8, // ’.’
    0x10,0x88,0x42,0x21,0x08,0x84,0x00, // ’/’
    0x78,0x92,0x14,0x28,0x50,0xA1,0x24,0x78, // ’0’
    0x62,0x82,0x08,0x20,0x82,0x08,0xF8, // ’1’
    0x79,0x08,0x10,0x20,0x82,0x08,0x22,0xFC, // ’2’
    0x79,0x08,0x10,0x23,0x81,0x81,0x42,0x78, // ’3’
    0x08,0x18,0x28,0x48,0x48,0xFE,0x08,0x08,0x1C, // ’4’
    0xF9,0x02,0x07,0xC8,0xC0,0x81,0x46,0x78, // ’5’
    0x38,0x8A,0x05,0xCC,0xD0,0xA1,0x26,0x78, // '6'
    0xFD,0x08,0x20,0x41,0x02,0x04,0x10,0x20, // '7'
    0x79,0x0A,0x14,0x27,0x99,0xA1,0x42,0x78, // '8'
    0x79,0x92,0x14,0x2C,0xCE,0x81,0x44,0x70, // '9'
    0xD8,0x6C, // ':'
    0xD8,0x04,0xA0, // ';'
    0x03,0x0E,0x38,0x1C,0x01,0xC0,0x18, // '<'
    0xFF,0x00,0x3F,0xC0, // '='
    0xC0,0x1C,0x01,0xC0,0xE3,0x86,0x00, // '>'
    0x73,0x28,0x86,0x30,0x80,0x0C,0x30, // '?'
    0x1E,0x04,0x31,0x03,0x47,0x29,0x25,0x24,0xA4,0xA4,0x78,0x40,0x0C,0x20,0x78,0x00, // '@'
    0x08,0x07,0x01,0x40,0x50,0x22,0x08,0x87,0xF1,0x04,0xE3,0x80, // 'A'
    0xFC,0x42,0x42,0x42,0x7C,0x42,0x42,0x42,0xFC, // 'B'
    0x3E,0x20,0xA0,0x10,0x08,0x04,0x02,0x00,0x86,0x3E,0x00, // 'C'
    0xFC,0x23,0x10,0x48,0x24,0x12,0x09,0x04,0x8C,0xFC,0x00, // 'D'
    0xFE,0x42,0x40,0x44,0x7C,0x44,0x40,0x42,0xFE, // 'E'
    0xFE,0x42,0x40,0x44,0x7C,0x44,0x40,0x40,0xE0, // 'F'
    0x3E,0x20,0xA0,0x10,0x08,0x04,0x3A,0x04,0x82,0x3E,0x00, // 'G'
    0xE3,0x90,0x44,0x11,0x04,0x7F,0x10,0x44,0x11,0x04,0xE3,0x80, // 'H'
    0xE4,0x44,0x44,0x44,0xE0, // 'I'
    0x38,0x41,0x04,0x10,0x41,0x04,0x10,0x49,0x38, // 'J'
    0xEE,0x22,0x12,0x0A,0x06,0x02,0x81,0x20,0x88,0xE3,0x00, // 'K'
    0xE0,0x40,0x40,0x40,0x40,0x40,0x40,0x42,0xFE, // 'L'
    0xE0,0xE6,0x0C,0x51,0x45,0x14,0x4A,0x44,0xA4,0x44,0x44,0x04,0xE0,0xE0, // 'M'
    0xC3,0x98,0x45,0x11,0x64,0x49,0x11,0x44,0x31,0x0C,0xE1,0x00, // 'N'
    0x3E,0x10,0x48,0x0A,0x02,0x80,0xA0,0x28,0x09,0x04,0x3E,0x00, // 'O'
    0xFC,0x46,0x42,0x46,0x7C,0x40,0x40,0x40,0xE0, // 'P'
    0x3E,0x10,0x48,0x0A,0x02,0x80,0xA0,0x28,0x09,0x04,0x3E,0x01,0x00,0x30, // 'Q'
    0xFC,0x10,0x84,0x21,0x08,0x7C,0x11,0x04,0x21,0x08,0xE1,0x80, // 'R'
    0x79,0x0A,0x06,0x03,0x01,0x81,0x42,0x78, // 'S'
    0xFE,0x92,0x10,0x10,0x10,0x10,0x10,0x10,0x38, // 'T'
    0xE3,0x90,0x44,0x11,0x04,0x41,0x10,0x44,0x11,0x8C,0x3E,0x00, // 'U'
    0xE3,0x90,0x44,0x30,0x88,0x22,0x05,0x01,0x40,0x70,0x08,0x00, // 'V'
    0xE6,0x72,0x31,0x11,0x88,0x52,0x82,0x94,0x14,0xA0,0xA5,0x02,0x10,0x10,0x80, // 'W'
    0xE7,0x21,0x09,0x03,0x01,0x80,0xC0,0x90,0x84,0xE7,0x00, // 'X'
    0xE3,0x90,0x42,0x20,0x50,0x08,0x02,0x00,0x80,0x20,0x1C,0x00, // 'Y'
    0xFE,0x84,0x0C,0x08,0x10,0x20,0x60,0x42,0xFE, // 'Z'
    0xE8,0x88,0x88,0x88,0x88,0xE0, // '['
    0x84,0x10,0x84,0x10,0x84,0x10,0x80, // '\'
    0xE2,0x22,0x22,0x22,0x22,0xE0, // ']'
    0x18,0x12,0x10,0x80, // '^'
    0xFC, // '_'
    0x42, // '`'
    0x71,0x10,0x23,0xC8,0x91,0x1F,0x00, // 'a'
    0xC0,0x40,0x40,0x5C,0x66,0x42,0x42,0x42,0x66,0xDC, // 'b'
    0x79,0x8A,0x04,0x08,0x18,0x9E,0x00, // 'c'
    0x0C,0x04,0x04,0x74,0xCC,0x84,0x84,0x84,0xCC,0x76, // 'd'
    0x79,0x92,0x17,0xE8,0x18,0x9E,0x00, // 'e'
    0x39,0x04,0x3C,0x41,0x04,0x10,0x43,0x80, // 'f'
    0x76,0xCC,0x84,0x84,0x84,0xCC,0x74,0x04,0x8C,0x78, // 'g'
    0xC0,0x40,0x40,0x58,0x64,0x44,0x44,0x44,0x44,0xEE, // 'h'
    0x40,0xC4,0x44,0x44,0xE0, // 'i'
    0x10,0x0C,0x21,0x08,0x42,0x10,0x85,0xE0, // 'j'
    0xC0,0x40,0x40,0x4C,0x48,0x50,0x70,0x48,0x44,0xEE, // 'k'
    0xC4,0x44,0x44,0x44,0x4E, // 'l'
    0xD9,0x86,0x64,0x44,0x44,0x44,0x44,0x44,0x44,0xEE,0xE0, // 'm'
    0xD8,0x64,0x44,0x44,0x44,0x44,0xEE, // 'n'
    0x79,0x9A,0x14,0x28,0x59,0x9E,0x00, // 'o'
    0xDC,0x66,0x42,0x42,0x42,0x66,0x5C,0x40,0x40,0xE0, // 'p'
    0x76,0xCC,0x84,0x84,0x84,0xCC,0x74,0x04,0x04,0x0E, // 'q'
    0xD9,0xA4,0x10,0x41,0x0E,0x00, // 'r'
    0x72,0x28,0x1C,0x0A,0x27,0x00, // 's'
    0x42,0x3C,0x84,0x21,0x0A,0x70, // 't'
    0xCC,0x44,0x44,0x44,0x44,0x44,0x3E, // 'u'
    0xEE,0x44,0x44,0x28,0x28,0x10,0x10, // 'v'
    0xE4,0xE4,0x44,0x4A,0x42,0xA8,0x2A,0x81,0x10,0x11,0x00, // 'w'
    0xEC,0x90,0xC1,0x83,0x09,0x37,0x00, // 'x'
    0xEE,0x44,0x44,0x68,0x28,0x28,0x10,0x10,0x30,0xE0, // 'y'
    0xFA,0x41,0x08,0x41,0x2F,0x80, // 'z'
    0x38,0x82,0x08,0x23,0x02,0x08,0x20,0x83,0x80, // '{'
    0xAA,0xAA,0xAA, // '|'
    0xE0,0x82,0x08,0x20,0x62,0x08,0x20,0x8E,0x00 // '}'
};
const GFXglyph Serif_plain_12Glyphs[] PROGMEM = {
// bitmapOffset, width, height, xAdvance, xOffset, yOffset
      {     0,   2,   1,   5,    0,   -1 }, // ' '
      {     1,   3,   9,   6,    2,   -9 }, // '!'
      {     5,   4,   3,   7,    1,   -9 }, // '"'
      {     7,   9,   9,  11,    1,   -9 }, // '#'
      {    18,   7,  12,   9,    1,  -10 }, // '$'
      {    29,  11,   9,  12,    1,   -9 }, // '%'
      {    42,  10,   9,  12,    1,   -9 }, // '&'
      {    54,   2,   3,   4,    1,   -9 }, // '''
      {    55,   4,  11,   6,    1,  -10 }, // '('
      {    61,   4,  11,   6,    1,  -10 }, // ')'
      {    67,   6,   5,   7,    1,   -9 }, // '*'
      {    71,   8,   7,  11,    2,   -7 }, // '+'
      {    78,   3,   3,   5,    1,   -1 }, // ’,’
      {    80,   4,   1,   5,    1,   -4 }, // ’-’
      {    81,   3,   2,   5,    1,   -2 }, // ’.’
      {    82,   5,  10,   5,    0,   -9 }, // ’/’
      {    89,   7,   9,   9,    1,   -9 }, // ’0’
      {    97,   6,   9,   9,    1,   -9 }, // ’1’
      {   104,   7,   9,   9,    1,   -9 }, // ’2’
      {   112,   7,   9,   9,    1,   -9 }, // ’3’
      {   120,   8,   9,   9,    1,   -9 }, // ’4’
      {   129,   7,   9,   9,    1,   -9 }, // ’5’
      {   137,   7,   9,   9,    1,   -9 }, // ’6’
      {   145,   7,   9,   9,    1,   -9 }, // ’7’
      {   153,   7,   9,   9,    1,   -9 }, // ’8’
      {   161,   7,   9,   9,    1,   -9 }, // ’9’
      {   169,   3,   5,   5,    1,   -5 }, // ’:’
      {   171,   3,   7,   5,    1,   -5 }, // ’;’
      {   174,   9,   6,  11,    1,   -7 }, // ’<’
      {   181,   9,   3,  11,    1,   -5 }, // ’=’
      {   185,   9,   6,  11,    1,   -7 }, // ’>’
      {   192,   6,   9,   7,    1,   -9 }, // ’?’
      {   199,  11,  11,  13,    1,   -8 }, // ’@’
      {   215,  10,   9,  10,    0,   -9 }, // ’A’
      {   227,   8,   9,  10,    1,   -9 }, // ’B’
      {   236,   9,   9,  10,    1,   -9 }, // ’C’
      {   247,   9,   9,  10,    1,   -9 }, // ’D’
      {   258,   8,   9,   9,    1,   -9 }, // ’E’
      {   267,   8,   9,   9,    1,   -9 }, // ’F’
      {   276,   9,   9,  10,    1,   -9 }, // ’G’
      {   287,  10,   9,  11,    1,   -9 }, // ’H’
      {   299,   4,   9,   5,    1,   -9 }, // ’I’
      {   304,   6,  12,   5,   -1,   -9 }, // ’J’
      {   313,   9,   9,  10,    1,   -9 }, // ’K’
      {   324,   8,   9,   9,    1,   -9 }, // ’L’
      {   333,  12,   9,  13,    1,   -9 }, // ’M’
      {   347,  10,   9,  11,    1,   -9 }, // ’N’
      {   359,  10,   9,  11,    1,   -9 }, // ’O’
      {   371,   8,   9,   9,    1,   -9 }, // ’P’
      {   380,  10,  11,  11,    1,   -9 }, // ’Q’
      {   394,  10,   9,  10,    1,   -9 }, // ’R’
      {   406,   7,   9,   8,    1,   -9 }, // ’S’
      {   414,   8,   9,   9,    1,   -9 }, // ’T’
      {   423,  10,   9,  11,    1,   -9 }, // ’U’
      {   435,  10,   9,  10,    0,   -9 }, // ’V’
      {   447,  13,   9,  13,    0,   -9 }, // ’W’
      {   462,   9,   9,  10,    1,   -9 }, // ’X’
      {   473,  10,   9,   9,    0,   -9 }, // ’Y’
      {   485,   8,   9,   9,    1,   -9 }, // ’Z’
      {   494,   4,  11,   6,    1,  -10 }, // ’[’
      {   500,   5,  10,   5,    0,   -9 }, // ’\’
      {   507,   4,  11,   6,    1,  -10 }, // ’]’
      {   513,   9,   3,  11,    1,   -9 }, // ’^’
      {   517,   7,   1,   7,    0,    2 }, // ’_’
      {   518,   4,   2,   7,    1,  -10 }, // ’`’
      {   519,   7,   7,   8,    1,   -7 }, // ’a’
      {   526,   8,  10,   9,    1,  -10 }, // ’b’
      {   536,   7,   7,   8,    1,   -7 }, // ’c’
      {   543,   8,  10,   9,    1,  -10 }, // ’d’
      {   553,   7,   7,   8,    1,   -7 }, // ’e’
      {   560,   6,  10,   6,    1,  -10 }, // ’f’
      {   568,   8,  10,   9,    1,   -7 }, // ’g’
      {   578,   8,  10,   9,    1,  -10 }, // ’h’
      {   588,   4,   9,   5,    1,   -9 }, // ’i’
      {   593,   5,  12,   5,   -1,   -9 }, // ’j’
      {   601,   8,  10,   9,    1,  -10 }, // ’k’
      {   611,   4,  10,   5,    1,  -10 }, // ’l’
      {   616,  12,   7,  13,    1,   -7 }, // ’m’
      {   627,   8,   7,   9,    1,   -7 }, // ’n’
      {   634,   7,   7,   8,    1,   -7 }, // ’o’
      {   641,   8,  10,   9,    1,   -7 }, // ’p’
      {   651,   8,  10,   9,    1,   -7 }, // ’q’
      {   661,   6,   7,   7,    1,   -7 }, // ’r’
      {   667,   6,   7,   7,    1,   -7 }, // ’s’
      {   673,   5,   9,   6,    1,   -9 }, // ’t’
      {   679,   8,   7,   9,    1,   -7 }, // ’u’
      {   686,   8,   7,   7,    0,   -7 }, // ’v’
      {   693,  12,   7,  11,    0,   -7 }, // ’w’
      {   704,   7,   7,   8,    1,   -7 }, // ’x’
      {   711,   8,  10,   8,    0,   -7 }, // ’y’
      {   721,   6,   7,   7,    1,   -7 }, // ’z’
      {   727,   6,  11,   9,    1,  -10 }, // ’{’
      {   736,   2,  12,   5,    2,   -9 }, // ’|’
      {   739,   6,  11,   9,    1,  -10 } // ’}’
};
const GFXfont Serif_plain_12 PROGMEM = {
(uint8_t  *)Serif_plain_12Bitmaps,(GFXglyph *)Serif_plain_12Glyphs,0x20, 0x7E, 15};
```

3.20 serif9.h

```
// Created by https://oleddisplay.squix.ch/ Consider a donation
// In case of problems make sure that you are using the font file with the correct version!
const uint8_t Serif_plain_9Bitmaps[] PROGMEM = {

    // Bitmap Data:
    0x00, // ' '
    0xAA,0x88, // '!'
    0xAA, // '"'
    0x28,0x51,0xF1,0x8F,0xCA,0x00, // '#'
    0x47,0xB1,0xC7,0x2B,0x88, // '$'
    0xE4,0x52,0x2A,0x1F,0xE1,0x51,0x28,0x9C, // '%'
    0x70,0x50,0x40,0xAE,0xB4,0x98,0x76, // '&'
    0xA0, // '''
    0x52,0x49,0x22, // '('
    0x89,0x24,0x94, // ')'
    0x23,0xEF,0x88, // '*'
    0x20,0x8F,0x88,0x20, // '+'
    0x50, // ','
    0xC0, // '-'
    0x80, // '.'
    0x22,0x44,0x48,0x80, // '/'
    0x72,0x28,0xA2,0x8A,0x27,0x00, // '0'
    0x4C,0x44,0x44,0xE0, // '1'
    0x64,0x84,0x44,0x2B,0xC0, // '2'
    0x64,0x84,0x41,0x49,0x80, // '3'
    0x10,0xC5,0x14,0xF8,0x43,0x80, // '4'
    0xF4,0x38,0x21,0x49,0x80, // '5'
    0x73,0x28,0x3C,0x8A,0x27,0x00, // '6'
    0xF4,0x88,0x42,0x21,0x00, // '7'
    0x72,0x28,0x9C,0xDA,0x27,0x00, // '8'
    0x72,0x28,0x9E,0x0A,0x67,0x00, // '9'
    0x82, // ':'
    0x40,0x28, // ';'
    0x04,0x73,0x03,0x80,0xC0, // '<'
    0xFC,0x03,0xF0, // '='
    0x80,0xE0,0x31,0xCC,0x00, // '>'
    0xE0,0x84,0xC4,0x01,0x00, // '?'
    0x3C,0x21,0x27,0x54,0xAA,0x54,0xF1,0x00,0x78, // '@'
    0x10,0x30,0x28,0x48,0x7C,0x44,0xC6, // 'A'
    0xF8,0x89,0x13,0xC4,0x48,0xBE,0x00, // 'B'
    0x38,0x8A,0x04,0x08,0x08,0x9E,0x00, // 'C'
    0xF8,0x99,0x12,0x24,0x49,0xBE,0x00, // 'D'
    0xFC,0x89,0x03,0xC4,0x08,0xBF,0x00, // 'E'
    0xFC,0x89,0x03,0xC4,0x08,0x38,0x00, // 'F'
    0x38,0x8A,0x04,0x08,0xD8,0x9E,0x00, // 'G'
    0xEE,0x44,0x44,0x7C,0x44,0x44,0xEE, // 'H'
    0xE4,0x44,0x44,0xE0, // 'I'
    0x71,0x08,0x42,0x10,0x94,0xC0, // 'J'
    0xEC,0x48,0x50,0x60,0x50,0x48,0xE6, // 'K'
    0xE1,0x04,0x10,0x41,0x2F,0x80, // 'L'
    0xE3,0x98,0xC5,0x51,0x54,0x49,0x12,0x4E,0x38, // 'M'
    0xCE,0x64,0x64,0x54,0x4C,0x4C,0xE4, // 'N'
    0x79,0x9A,0x14,0x28,0x59,0x9E,0x00, // 'O'
    0xF1,0x24,0x9C,0x41,0x0E,0x00, // 'P'
    0x79,0x9A,0x14,0x28,0x59,0x1E,0x04, // 'Q'
    0xF8,0x91,0x23,0x85,0x09,0x39,0x00, // 'R'
    0x72,0x28,0x1C,0x0A,0x27,0x00, // 'S'
    0xFA,0xA2,0x08,0x20,0x87,0x00, // 'T'
    0xEE,0x44,0x44,0x44,0x44,0x44,0x38, // 'U'
    0xCE,0x44,0x44,0x28,0x28,0x10,0x10, // 'V'
    0xEB,0x92,0x45,0x51,0x54,0x36,0x08,0x82,0x20, // 'W'
    0xDC,0x90,0xC1,0x83,0x09,0x3B,0x00, // 'X'
    0xEE,0x44,0x28,0x10,0x10,0x10,0x38, // 'Y'
    0xFA,0x21,0x08,0x42,0x2F,0x80, // 'Z'
    0xD2,0x49,0x26, // '['
    0x88,0x44,0x42,0x20, // '\'
    0xC9,0x24,0x96, // ']'
    0x30,0x90, // '^'
    0xF8, // '_'
    0x88, // '`'
    0x60,0x4F,0x24,0xF8, // 'a'
    0xC1,0x04,0x1C,0x49,0x24,0xBC, // 'b'
    0x64,0xA1,0x07,0x00, // 'c'
    0x30,0x41,0x1C,0x92,0x49,0x1E, // 'd'
    0x64,0xBD,0x07,0x00, // 'e'
    0x72,0x11,0xC4,0x21,0x1C, // 'f'
    0x7A,0x49,0x24,0x70,0x4E,0x00, // 'g'
    0xC0,0x81,0x03,0xC4,0x89,0x12,0x76, // 'h'
    0x40,0xC4,0x44,0xE0, // 'i'
    0x20,0x62,0x22,0x22,0xE0, // 'j'
    0xC0,0x81,0x02,0xC5,0x0E,0x12,0x76, // 'k'
    0xC4,0x44,0x44,0x4E, // 'l'
    0xFF,0x12,0x44,0x91,0x24,0xED,0x80, // 'm'
    0xF8,0x91,0x22,0x4E,0xC0, // 'n'
    0x64,0xA5,0x26,0x00, // 'o'
    0xF1,0x24,0x92,0x71,0x0E,0x00, // 'p'
    0x7A,0x49,0x24,0x70,0x43,0x80, // 'q'
    0xF2,0x90,0x8E,0x00, // 'r'
    0xF4,0x18,0x2F,0x00, // 's'
    0x47,0x10,0x85,0x38, // 't'
    0xD8,0x91,0x22,0x47,0xC0, // 'u'
    0xD9,0x45,0x08,0x20, // 'v'
    0xD6,0x54,0x54,0x28,0x28, // 'w'
    0xD9,0x42,0x14,0xD8, // 'x'
    0xD9,0x45,0x08,0x20,0x8C,0x00, // 'y'
    0xF5,0x18,0xAF,0x00, // 'z'
    0x64,0x44,0x84,0x46, // '{'
    0xAA,0xAA,0x80, // '|'
    0xC4,0x44,0x24,0x4C // '}'
};
const GFXglyph Serif_plain_9Glyphs[] PROGMEM = {
// bitmapOffset, width, height, xAdvance, xOffset, yOffset
      {     0,   2,   1,   4,    0,   -1 }, // ' '
      {     1,   2,   7,   5,    1,   -7 }, // '!'
      {     3,   4,   2,   5,    1,   -7 }, // '"'
      {     4,   7,   6,   9,    0,   -6 }, // '#'
      {    10,   5,   8,   7,    0,   -7 }, // '$'
      {    15,   9,   7,  10,    0,   -7 }, // '%'
      {    23,   8,   7,   9,    0,   -7 }, // '&'
      {    30,   2,   2,   3,    1,   -7 }, // '''
      {    31,   3,   8,   5,    0,   -8 }, // '('
      {    34,   3,   8,   5,    1,   -8 }, // ')'
      {    37,   6,   4,   6,    0,   -7 }, // '*'
      {    40,   6,   5,   9,    1,   -5 }, // '+'
      {    44,   3,   2,   4,    0,   -1 }, // ','
      {    45,   3,   1,   4,    0,   -3 }, // '-'
      {    46,   2,   1,   4,    1,   -1 }, // '.'
      {    47,   4,   7,   4,    0,   -7 }, // '/'
      {    51,   6,   7,   7,    0,   -7 }, // '0'
      {    57,   4,   7,   7,    1,   -7 }, // '1'
      {    61,   5,   7,   7,    0,   -7 }, // '2'
      {    66,   5,   7,   7,    0,   -7 }, // '3'
      {    71,   6,   7,   7,    0,   -7 }, // '4'
      {    77,   5,   7,   7,    1,   -7 }, // '5'
      {    82,   6,   7,   7,    0,   -7 }, // '6'
      {    88,   5,   7,   7,    1,   -7 }, // '7'
      {    93,   6,   7,   7,    0,   -7 }, // '8'
      {    99,   6,   7,   7,    0,   -7 }, // '9'
      {   105,   2,   4,   4,    1,   -4 }, // ':'
      {   106,   3,   5,   4,    0,   -4 }, // ';'
      {   108,   7,   5,   9,    0,   -5 }, // '<'
      {   113,   7,   3,   9,    0,   -4 }, // '='
      {   116,   7,   5,   9,    0,   -5 }, // '>'
      {   121,   5,   7,   6,    0,   -7 }, // '?'
      {   126,   9,   8,  10,    0,   -6 }, // '@'
      {   135,   8,   7,   8,    0,   -7 }, // 'A'
      {   142,   7,   7,   8,    0,   -7 }, // 'B'
      {   149,   7,   7,   8,    0,   -7 }, // 'C'
      {   156,   7,   7,   8,    0,   -7 }, // 'D'
      {   163,   7,   7,   8,    0,   -7 }, // 'E'
      {   170,   7,   7,   7,    0,   -7 }, // 'F'
      {   177,   7,   7,   8,    0,   -7 }, // 'G'
      {   184,   8,   7,   9,    0,   -7 }, // 'H'
      {   191,   4,   7,   5,    0,   -7 }, // 'I'
      {   195,   5,   9,   5,   -1,   -7 }, // 'J'
      {   201,   8,   7,   8,    0,   -7 }, // 'K'
      {   208,   6,   7,   7,    0,   -7 }, // 'L'
      {   214,  10,   7,  11,    0,   -7 }, // 'M'
      {   223,   8,   7,   9,    0,   -7 }, // 'N'
      {   230,   7,   7,   8,    0,   -7 }, // 'O'
      {   237,   6,   7,   7,    0,   -7 }, // 'P'
      {   243,   7,   8,   8,    0,   -7 }, // 'Q'
      {   250,   7,   7,   8,    0,   -7 }, // 'R'
      {   257,   6,   7,   7,    0,   -7 }, // 'S'
      {   263,   6,   7,   7,    0,   -7 }, // 'T'
      {   269,   8,   7,   9,    0,   -7 }, // 'U'
      {   276,   8,   7,   8,    0,   -7 }, // 'V'
      {   283,  10,   7,  10,    0,   -7 }, // 'W'
      {   292,   7,   7,   8,    0,   -7 }, // 'X'
      {   299,   8,   7,   7,   -1,   -7 }, // 'Y'
      {   306,   6,   7,   7,    0,   -7 }, // 'Z'
      {   312,   3,   8,   5,    0,   -8 }, // '['
      {   315,   4,   7,   4,    0,   -7 }, // '\'
      {   319,   3,   8,   5,    1,   -8 }, // ']'
      {   322,   7,   2,   9,    1,   -7 }, // '^'
      {   324,   6,   1,   6,    0,    1 }, // '_'
      {   325,   3,   2,   6,    1,   -8 }, // '`'
      {   326,   6,   5,   6,    0,   -5 }, // 'a'
      {   330,   6,   8,   7,    0,   -8 }, // 'b'
      {   336,   5,   5,   6,    0,   -5 }, // 'c'
      {   340,   6,   8,   6,    0,   -8 }, // 'd'
      {   346,   5,   5,   6,    0,   -5 }, // 'e'
      {   350,   5,   8,   4,    0,   -8 }, // 'f'
      {   355,   6,   7,   6,    0,   -5 }, // 'g'
      {   361,   7,   8,   7,    0,   -8 }, // 'h'
      {   368,   4,   7,   4,    0,   -7 }, // 'i'
      {   372,   4,   9,   4,   -1,   -7 }, // 'j'
      {   377,   7,   8,   7,    0,   -8 }, // 'k'
      {   384,   4,   8,   4,    0,   -8 }, // 'l'
      {   388,  10,   5,  10,    0,   -5 }, // 'm'
      {   395,   7,   5,   7,    0,   -5 }, // 'n'
      {   400,   5,   5,   6,    0,   -5 }, // 'o'
      {   404,   6,   7,   7,    0,   -5 }, // 'p'
      {   410,   6,   7,   7,    0,   -5 }, // 'q'
      {   416,   5,   5,   5,    0,   -5 }, // 'r'
      {   420,   5,   5,   6,    0,   -5 }, // 's'
      {   424,   5,   6,   5,    0,   -6 }, // 't'
      {   428,   7,   5,   7,    0,   -5 }, // 'u'
      {   433,   6,   5,   6,    0,   -5 }, // 'v'
      {   437,   8,   5,   9,    0,   -5 }, // 'w'
      {   442,   6,   5,   6,    0,   -5 }, // 'x'
      {   446,   6,   7,   6,    0,   -5 }, // 'y'
      {   452,   5,   5,   6,    0,   -5 }, // 'z'
      {   456,   4,   8,   7,    1,   -8 }, // '{'
      {   460,   2,   9,   4,    1,   -7 }, // '|'
      {   463,   4,   8,   7,    1,   -8 } // '}'
};
const GFXfont Serif_plain_9 PROGMEM = {
(uint8_t  *)Serif_plain_9Bitmaps,(GFXglyph *)Serif_plain_9Glyphs,0x20, 0x7E, 12};
```

3.21 globals/globals.cpp File Reference

This file contains global variables, constants, and configurations used throughout the project.

```
#include "globals.h"
#include <libraries.h>
```

Functions

- ThreeWire **myWire** (21, 22, 4)
- IPAddress **ip** (192, 168, 1, 200)
- RtcDS1302< ThreeWire > **Rtc** (myWire)
- BfButton **btn** (BfButton::STANDALONE_DIGITAL, 5, true, LOW)
- EthernetServer **server** (80)
- AccelStepper **motor** (AccelStepper::DRIVER, 2, 15)

Variables

- const uint32_t LOOP_TICKS = 50 / portTICK_PERIOD_MS

  Timing and Scheduling Constants.
- const unsigned long **min_bright_timer** = 180000

  Minimum time before dimming the screen (in milliseconds)
- const unsigned long **turn_off_timer** = 600000

  Timer for turning off the display (in milliseconds)
- const int DIR_PIN = 15

  Rotary Encoder Pin Definitions.
- const int **STEP_PIN** = 2

  Step pin for rotary encoder.
- int **DT** = 16

  Data pin for rotary encoder.
- int **CLK** = 17

Clock pin for rotary encoder.

- int lastClk = HIGH

  Rotary Encoder State Variables.
- int **newClk** = digitalRead(CLK)

  Current clock state.
- int **dtValue** = digitalRead(DT)

  Data pin value for rotary encoder.
- int **counter** = 0

  Encoder counter to track position.
- int **angle** = 0

  Current angle of the encoder.
- int **aState**

  Current state of the encoder.
- int **aLastState**

  Last state of the encoder.
- long **position** = 0

  Rotary encoder position.
- int **lastAngle** = 0

  Last measured angle.
- int centerX = 80

  Display Geometry Variables.
- int **centerY** = 74

  Y coordinate for center of display.
- int **radius** = 40

  Radius of UI elements.
- const int minBrightness = 200

  Brightness Control Variables.
- const int **maxBrightness** = 230

  Maximum brightness level.
- const int **numBrightnessLevels** = 10

  Total number of brightness levels.
- int **brightnessLevel** = 5

  Current brightness level.
- MenuState currentMenu = INITIAL_SCREEN

  MenuState enum values represent different menu screens.
- MenuState **previousMenu** = MAIN_MENU

  Previous menu state.
- int menuIndex = 0

  Menu Structure and Indexes.
- const int **menuItems** = 4

  Total number of main menu items.
- String **menu** [menuItems] = {"MAIN MENU","Settings","TCP server","Stepper motor"}

  Main menu labels.
- const int **submenuItems** = 4

  Total number of submenu items.
- String **submenu** [submenuItems] = {"SETTINGS","Brightness","Colors","Font type"}

  Submenu labels.
- const int **colormenuItems** = 5

  Number of items in color menu.
- String **colormenu** [colormenuItems] = {"COLORS","Background","Selection bar","Selection border","Text"}

  Color menu labels.

- const int **ethernetmenuItems** = 5
- String **ethernetmenu** [ethernetmenuItems] = {"IP","Port","Subnet","Status","Back"}
- const int **steppermenuItems** = 3
- String **steppermenu** [steppermenuItems] = {"STEPPER","Automatic", "Manual"}
- const int **manualmenuItems** = 3
- String **manualmenu** [manualmenuItems] = {"MANUAL","Forward","Backward"}
- const int **automenuItems** = 1
- String **automenu** [automenuItems] = {"AUTO"}
- byte **mac** [ ] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }

  Ethernet Configuration.
- char **ipchar** [16]
- char **subnetchar** [16]
- IPAddress **subnet** = Ethernet.subnetMask()
- char **ipadress** = sprintf(ipchar, "%d.%d.%d.%d", ip[0],ip[1],ip[2],ip[3])
- char **subnetadress** = sprintf(subnetchar, "%d.%d.%d.%d", subnet[0],subnet[1],subnet[2],subnet[3])
- String **ipip** = ipchar
- String **subnetsubnet** = subnetchar
- String **ethernetInfo** [5] = {ipip, "80", subnetsubnet, ""}

  Ethernet settings: IP, Port, Subnet, Status.
- const unsigned short ∗ images_main [ ] = {NULL, settings_icon, ethernet_icon, step_icon}

  Icons for Menus.
- const unsigned short ∗ **images_settings** [ ] = {NULL, brightness_icon, color_icon, font_icon}

  Settings menu icons.
- const unsigned short ∗ **images_stepper** [ ] = {NULL, automatic_icon, manual_icon}

  Stepper motor menu icons.
- bool screenOn = true

  Screen Control Variables.
- bool **engineeringMode** = false

  Flag for enabling engineering/debug mode.
- volatile int **encoderState** = 0

  Encoder state for handling inputs.
- RtcDateTime **lastDisplayedTime**

  Time last displayed on the screen.
- QueueHandle_t screenUpdateQueue

  Queues and Synchronization.
- xSemaphoreHandle **gatekeeper** = 0

  Semaphore for managing screen updates.
- const uint16_t backgroundpossibleColors [ ] = {MINT_GREEN, TFT_WHITE, LIGHT_BLUE, PALE_GRAY, PALE_YELLOW, PALE_PINK, BEIGE, LAVENDER, SKY_BLUE}

  Color Arrays for Background, Selection Bar, Border, and Text.
- const uint16_t **selectbarpossibleColors** [ ] = {TFT_RED, TFT_BLUE, TFT_YELLOW, TFT_PINK, TFT_↩ ORANGE, TFT_CYAN, TFT_SKYBLUE, TFT_VIOLET}

  Possible selection bar colors.
- const uint16_t **borderpossibleColors** [ ] = {TFT_BLACK, TFT_RED, TFT_GREENYELLOW, TFT_BLUE, TFT_YELLOW, TFT_PINK, TFT_ORANGE, TFT_CYAN, TFT_LIGHTGREY, TFT_SKYBLUE, TFT_VIOLET}

  Possible border colors.
- const uint16_t **textpossibleColors** [ ] = {TFT_BLACK, DARK_ORANGE, DARK_BLUE, DARK_GREEN}

  Possible text colors.
- const int backgroundcolors = 9

  Color Count Variables.
- const int **selectbarcolors** = 8

  Number of selection bar colors.

- const int **textcolors** = 4

  Number of text colors.
- const int **bordercolors** = 11

  Number of border colors.
- const int **selectnumcolors** [ ] ={backgroundcolors,selectbarcolors,textcolors,bordercolors}
- MenuState colormenuArray [ ] = {MenuState::SELECTION_BAR, MenuState::BACKGROUND, MenuState::TEXT, MenuState::SELECTION_BORDER}

  Color Menu Arrays.
- String **background** [backgroundcolors] = {"Mint","White","Light blue", "Pale gray", "Yellow","Pale pink","Beige", "Lavender","Skyblue"}

  Background color labels.
- String **selectionbar** [selectbarcolors] = {"Red", "Blue", "Yellow","Pink", "Orange","Cyan", "Skyblue","Violet"}

  Selection bar color labels.
- String **border** [bordercolors] = {"Black","Red","Green","Blue","Yellow","Pink","Orange","Cyan","Light grey","Skyblue","Violet"}

  Border color labels.
- String **text** [textcolors] = {"Black","Orange","Dark blue","Dark green"}

  Text color labels.
- int selectedBackgroundColor = 1

  Selected Colors.
- int **selectedTextColor** = 0

  Selected text color index.
- int **selectedBarColor** = 5

  Selected selection bar color index.
- int **selectedBorderColor** = 0

  Selected border color index.
- int ∗ **selection** [ ] = {NULL, &selectedBackgroundColor, &selectedBarColor, &selectedBorderColor, &selectedTextColor}

  Array for selected colors.
- const GFXfont ∗ possibleFont12 [ ] = {NULL, sansserif12, mono12, arimo12, serif12}

  Font Configuration.
- const GFXfont ∗ **possibleFont9** [ ] = {NULL, sansserif9, mono9, arimo9, serif9}

  Array of possible fonts (9-point size)
- const int **numFonts** = sizeof(possibleFont12) / sizeof(possibleFont12[0])

  Number of font options.
- String **fontType** [numFonts] = {"TEXT FONT", "SansSerif", "Monospaced", "Arimo", "Serif"}

  Font type labels.
- const GFXfont ∗ **textType12** = possibleFont12[4]

  Selected 12-point font.
- const GFXfont ∗ **textType9** = possibleFont9[4]

  Selected 9-point font.
- TFT_eSPI **tft** = TFT_eSPI()

  Hardware Initialization.
- TickType_t **previousTick** = 0
- TickType_t **previousTime** = 0

3.21.1 Detailed Description

This file contains global variables, constants, and configurations used throughout the project.

Author

Juan Alberto Serrano Redondo.

3.21.2 Variable Documentation

3.21.2.1 backgroundcolors

```
const int backgroundcolors = 9
```

Color Count Variables.

Number of background colors.

Number of background colors

3.21.2.2 backgroundpossibleColors

```
const uint16_t backgroundpossibleColors[] = {MINT_GREEN, TFT_WHITE, LIGHT_BLUE, PALE_GRAY,
PALE_YELLOW, PALE_PINK, BEIGE, LAVENDER, SKY_BLUE}
```

Color Arrays for Background, Selection Bar, Border, and Text.

Possible background colors.

Possible background colors

3.21.2.3 centerX

```
int centerX = 80
```

Display Geometry Variables.

X coordinate for center of display.

X coordinate for center of display

3.21.2.4 colormenuArray

```
MenuState colormenuArray[] = {MenuState::SELECTION_BAR, MenuState::BACKGROUND, MenuState::TEXT,
MenuState::SELECTION_BORDER}
```

Color Menu Arrays.

Color menu states.

Color menu states

3.21.2.5 currentMenu

```
MenuState currentMenu = INITIAL_SCREEN
```

MenuState enum values represent different menu screens.

Current menu state.

Current menu state

3.21.2.6 DIR_PIN

```
const int DIR_PIN = 15
```

Rotary Encoder Pin Definitions.

Direction pin for rotary encoder.

Direction pin for rotary encoder

3.21.2.7 images_main

```
const unsigned short* images_main[ ] = {NULL, settings_icon, ethernet_icon, step_icon}
```

Icons for Menus.

Main menu icons.

Main menu icons

3.21.2.8 lastClk

```
int lastClk = HIGH
```

Rotary Encoder State Variables.

Last clock state.

Last clock state

3.21.2.9 LOOP_TICKS

```
const uint32_t LOOP_TICKS = 50 / portTICK_PERIOD_MS
```

Timing and Scheduling Constants.

Loop period in ticks for RTOS task.

Loop period in ticks for RTOS task

3.21.2.10 menuIndex

```
int menuIndex = 0
```

Menu Structure and Indexes.

Index for current menu item.

Index for current menu item

3.21.2.11 minBrightness

```
const int minBrightness = 200
```

Brightness Control Variables.

Minimum brightness level.

Minimum brightness level

3.21.2.12 possibleFont12

```
const GFXfont* possibleFont12[ ] = {NULL, sansserif12, mono12, arimo12, serif12}
```

Font Configuration.

Array of possible fonts (12-point size)

3.21.2.13 screenOn

```
bool screenOn = true
```

Screen Control Variables.

Flag indicating if the screen is currently on.

Flag indicating if the screen is currently on

3.21.2.14 screenUpdateQueue

```
QueueHandle_t screenUpdateQueue
```

Queues and Synchronization.

Queue for handling screen update commands.

Queue for handling screen update commands

3.21.2.15 selectedBackgroundColor

```
int selectedBackgroundColor = 1
```

Selected Colors.

Selected background color index.

Selected background color index

3.22 globals/globals.h File Reference

Global variable and constant definitions for the project.

```
#include <libraries.h>
```

Macros

- #define **LOAD_GFXFF**
- #define **BRIGHTNESS_PIN** 25

  Pin for brightness control.
- #define **LIGHT_GRAY** 0xC618

  Light gray color for UI elements.
- #define **LIGHT_BLUE** 0xE7DF

  Light blue color for UI elements.
- #define **PALE_GRAY** 0xE71C

  Pale gray color for UI elements.
- #define **MINT_GREEN** 0x97F6

  Mint green color for UI elements.
- #define **PALE_YELLOW** 0xFFFB

  Pale yellow color for UI elements.
- #define **PALE_PINK** 0xFDBC

  Pale pink color for UI elements.
- #define **BEIGE** 0xFF7B

  Beige color for UI elements.
- #define **LAVENDER** 0xE5FF

  Lavender color for UI elements.
- #define **SKY_BLUE** 0x9FFF

  Sky blue color for UI elements.
- #define **DARK_ORANGE** 0xDBC9

  Dark orange color for UI elements.
- #define **DARK_BLUE** 0x00B1

  Dark blue color for UI elements.
- #define **DARK_GREEN** 0x22A2

  Dark green color for UI elements.

Enumerations

- enum MenuState {
  MAIN_MENU , SETTINGS , BRIGHTNESS , COLOR ,
  TEXT_TYPE , BACKGROUND , SELECTION_BAR , SELECTION_BORDER ,
  TEXT , SERVIDOR_TCP , STEPPER_MOTOR , MANUAL ,
  AUTO , INITIAL_SCREEN }

  Defines various states of the menu interface.
- enum ScreenUpdateCommand { UPDATE_MAIN_MENU , UPDATE_CONFIG_BRIGHTNESS , UPDATE_CONFIG_BACKGRO
  , UPDATE_CONFIG_ASPECT }

  Enum for screen update commands.

Variables

- const uint32_t LOOP_TICKS

  Loop period in ticks for RTOS task.
- const unsigned long **min_bright_timer**

  Minimum time before dimming the screen (in milliseconds)
- const unsigned long **turn_off_timer**

  Timer for turning off the display (in milliseconds)
- const int DIR_PIN

  Direction pin for rotary encoder.
- const int **STEP_PIN**

  Step pin for rotary encoder.
- int **DT**

  Data pin for rotary encoder.
- int **CLK**

  Clock pin for rotary encoder.
- int lastClk

  Last clock state.
- int **newClk**

  Current clock state.
- int **dtValue**

  Data pin value for rotary encoder.
- int **counter**

  Encoder counter to track position.
- int **angle**

  Current angle of the encoder.
- int **aState**

  Current state of the encoder.
- int **aLastState**

  Last state of the encoder.
- long **position**

  Rotary encoder position.
- int **lastAngle**

  Last measured angle.
- int centerX

  X coordinate for center of display.
- int **centerY**

  Y coordinate for center of display.
- int **radius**

  Radius of UI elements.
- const int minBrightness

  Minimum brightness level.
- const int **maxBrightness**

  Maximum brightness level.
- const int **numBrightnessLevels**

  Total number of brightness levels.
- int **brightnessLevel**

  Current brightness level.
- MenuState currentMenu

  Current menu state.
- MenuState **previousMenu**

Previous menu state.
- int menuIndex

  Index for current menu item.
- const int **menuItems**

  Total number of main menu items.
- String **menu** [ ]

  Main menu labels.
- const int **submenuItems**

  Total number of submenu items.
- String **submenu** [ ]

  Submenu labels.
- const int **colormenuItems**

  Number of items in color menu.
- String **colormenu** [ ]

  Color menu labels.
- const int **ethernetmenuItems**
- String **ethernetmenu** [ ]
- const int **steppermenuItems**
- String **steppermenu** [ ]
- const int **manualmenuItems**
- String **manualmenu** [ ]
- const int **automenuItems**
- String **automenu** [ ]
- byte **mac** [ ]

  Ethernet Configuration.
- IPAddress **ip**
- EthernetServer **server**
- char **ipchar** [16]
- char **subnetchar** [16]
- IPAddress **subnet**
- char **ipadress**
- char **subnetadress**
- String **ipip**
- String **subnetsubnet**
- String **ethernetInfo** [5]

  Ethernet settings: IP, Port, Subnet, Status.
- const unsigned short ∗ images_main [ ]

  Main menu icons.
- const unsigned short ∗ **images_settings** [ ]

  Settings menu icons.
- const unsigned short ∗ **images_stepper** [ ]

  Stepper motor menu icons.
- bool screenOn

  Flag indicating if the screen is currently on.
- bool **engineeringMode**

  Flag for enabling engineering/debug mode.
- volatile int **encoderState**

  Encoder state for handling inputs.
- RtcDateTime **lastDisplayedTime**

  Time last displayed on the screen.
- QueueHandle_t screenUpdateQueue

  Queue for handling screen update commands.

- xSemaphoreHandle **gatekeeper**

    Semaphore for managing screen updates.
- const uint16_t backgroundpossibleColors [ ]

    Possible background colors.
- const uint16_t **selectbarpossibleColors** [ ]

    Possible selection bar colors.
- const uint16_t **borderpossibleColors** [ ]

    Possible border colors.
- const uint16_t **textpossibleColors** [ ]

    Possible text colors.
- const int backgroundcolors

    Number of background colors.
- const int **selectbarcolors**

    Number of selection bar colors.
- const int **textcolors**

    Number of text colors.
- const int **bordercolors**

    Number of border colors.
- const int **selectnumcolors** [ ]
- MenuState colormenuArray [ ]

    Color menu states.
- String **background** [ ]

    Background color labels.
- String **selectionbar** [ ]

    Selection bar color labels.
- String **border** [ ]

    Border color labels.
- String **text** [ ]

    Text color labels.
- int selectedBackgroundColor

    Selected background color index.
- int **selectedTextColor**

    Selected text color index.
- int **selectedBarColor**

    Selected selection bar color index.
- int **selectedBorderColor**

    Selected border color index.
- int ∗ **selection** [ ]

    Array for selected colors.
- const GFXfont ∗ possibleFont12 [ ]

    Array of possible fonts (12-point size)
- const GFXfont ∗ **possibleFont9** [ ]

    Array of possible fonts (9-point size)
- const int **numFonts**

    Number of font options.
- String **fontType** [ ]

    Font type labels.
- const GFXfont ∗ **textType12**

    Selected 12-point font.
- const GFXfont ∗ **textType9**

    Selected 9-point font.

- BfButton **btn**
- ThreeWire **myWire**
- RtcDS1302< ThreeWire > **Rtc**
- AccelStepper **motor**
- TFT_eSPI **tft**

  Hardware Initialization.
- TickType_t **previousTick**
- TickType_t **previousTime**

3.22.1 Detailed Description

Global variable and constant definitions for the project.

This header file contains the global definitions for UI colors, menu states, rotary encoder configurations, display settings, and other system-wide constants and variables. It also declares external variables and enums that are used across different modules of the program, ensuring centralized configuration and ease of modification.

The file includes:

- Custom color definitions for user interface (UI) elements.
- Pin assignments for hardware components (brightness control, rotary encoder).
- Timing constants and menu states (main menu, settings, motor control).
- Rotary encoder state tracking and UI geometry settings.
- Brightness control and menu structure.
- Ethernet configuration parameters and TCP server settings.
- Screen control and synchronization using FreeRTOS tasks and queues.
- Enum definitions for menus and screen updates.

Author

Juan Alberto Serrano Redondo.

3.22.2 Enumeration Type Documentation

3.22.2.1 MenuState

```
enum MenuState
```

Defines various states of the menu interface.

Enumerator

MAIN_MENU	Main menu.
SETTINGS	Settings menu.
BRIGHTNESS	Brightness control menu.
COLOR	Color configuration menu.
TEXT_TYPE	Text font selection.

Enumerator

BACKGROUND	Background color selection.
SELECTION_BAR	Selection bar color.
SELECTION_BORDER	Selection border color.
TEXT	Text color configuration.
SERVIDOR_TCP	TCP server configuration.
STEPPER_MOTOR	Stepper motor control.
MANUAL	Manual stepper control.
AUTO	Automatic stepper control.
INITIAL_SCREEN	Initial splash screen.

3.22.2.2 ScreenUpdateCommand

```
enum ScreenUpdateCommand
```

Enum for screen update commands.

Enumerator

UPDATE_MAIN_MENU	Command to update the main menu.
UPDATE_CONFIG_BRIGHTNESS	Command to update brightness configuration.
UPDATE_CONFIG_BACKGROUNDCOLOR	Command to update background color.
UPDATE_CONFIG_ASPECT	Command to update UI aspects.

3.22.3 Variable Documentation

3.22.3.1 backgroundcolors

```
const int backgroundcolors  [extern]
```

Number of background colors.

Number of background colors.

Number of background colors

3.22.3.2 backgroundpossibleColors

```
const uint16_t backgroundpossibleColors[]  [extern]
```

Possible background colors.

Possible background colors.

Possible background colors

3.22.3.3 centerX

```
int centerX [extern]
```

X coordinate for center of display.

X coordinate for center of display.

X coordinate for center of display

3.22.3.4 colormenuArray

```
MenuState colormenuArray[ ] [extern]
```

Color menu states.

Color menu states.

Color menu states

3.22.3.5 currentMenu

```
MenuState currentMenu [extern]
```

Current menu state.

Current menu state.

Current menu state

3.22.3.6 DIR_PIN

```
const int DIR_PIN [extern]
```

Direction pin for rotary encoder.

Direction pin for rotary encoder.

Direction pin for rotary encoder

3.22.3.7 images_main

```
const unsigned short* images_main[ ] [extern]
```

Main menu icons.

Main menu icons.

Main menu icons

3.22.3.8 lastClk

```
int lastClk [extern]
```

Last clock state.

Last clock state.

Last clock state

3.22.3.9 LOOP_TICKS

```
const uint32_t LOOP_TICKS [extern]
```

Loop period in ticks for RTOS task.

Loop period in ticks for RTOS task.

Loop period in ticks for RTOS task

3.22.3.10 menuIndex

```
int menuIndex [extern]
```

Index for current menu item.

Index for current menu item.

Index for current menu item

3.22.3.11 minBrightness

```
const int minBrightness [extern]
```

Minimum brightness level.

Minimum brightness level.

Minimum brightness level

3.22.3.12 possibleFont12

```
const GFXfont* possibleFont12[] [extern]
```

Array of possible fonts (12-point size)

Array of possible fonts (12-point size)

3.22.3.13 screenOn

```
bool screenOn  [extern]
```

Flag indicating if the screen is currently on.

Flag indicating if the screen is currently on.

Flag indicating if the screen is currently on

3.22.3.14 screenUpdateQueue

```
QueueHandle_t screenUpdateQueue  [extern]
```

Queue for handling screen update commands.

Queue for handling screen update commands.

Queue for handling screen update commands

3.22.3.15 selectedBackgroundColor

```
int selectedBackgroundColor  [extern]
```

Selected background color index.

Selected background color index.

Selected background color index

3.23 globals.h

Go to the documentation of this file.

```

#ifndef GLOBALS_H
#define GLOBALS_H

#include <libraries.h>


// Graphics and Pin Definitions
#define LOAD_GFXFF
#define BRIGHTNESS_PIN 25

// Custom Color Definitions
#define LIGHT_GRAY 0xC618
#define LIGHT_BLUE 0xE7DF
#define PALE_GRAY 0xE71C
#define MINT_GREEN 0x97F6
#define PALE_YELLOW 0xFFFB
#define PALE_PINK 0xFDBC
#define BEIGE 0xFF7B
#define LAVENDER 0xE5FF
#define SKY_BLUE 0x9FFF
#define DARK_ORANGE 0xDBC9
#define DARK_BLUE 0x00B1
#define DARK_GREEN 0x22A2

// Timing and Scheduling Constants
extern const uint32_t LOOP_TICKS;
extern const unsigned long min_bright_timer;
extern const unsigned long turn_off_timer;
```

```

// Rotary Encoder Pin Definitions
extern const int DIR_PIN;
extern const int STEP_PIN;
extern int DT;
extern int CLK;

// Rotary Encoder State Variables
extern int lastClk;
extern int newClk;
extern int dtValue;
extern int counter;
extern int angle;
extern int aState;
extern int aLastState;
extern long position;
extern int lastAngle;

// Display Geometry Variables
extern int centerX;
extern int centerY;
extern int radius;

// Brightness Control Variables
extern const int minBrightness;
extern const int maxBrightness;
extern const int numBrightnessLevels;
extern int brightnessLevel;

// Enum for Menu States
enum MenuState {
  MAIN_MENU,
  SETTINGS,
  BRIGHTNESS,
  COLOR,
  TEXT_TYPE,
  BACKGROUND,
  SELECTION_BAR,
  SELECTION_BORDER,
  TEXT,
  SERVIDOR_TCP,
  STEPPER_MOTOR,
  MANUAL,
  AUTO,
  INITIAL_SCREEN
};

extern MenuState currentMenu;
extern MenuState previousMenu;

// Menu Structure and Indexes
extern int menuIndex;
extern const int menuItems;
extern String menu[];
extern const int submenuItems;
extern String submenu[];
extern const int colormenuItems;
extern String colormenu[];
extern const int ethernetmenuItems;
extern String ethernetmenu[];
extern const int steppermenuItems;
extern String steppermenu[];
extern const int manualmenuItems;
extern String manualmenu[];
extern const int automenuItems;
extern String automenu[];
// Ethernet Configuration
extern byte mac[];
extern IPAddress ip;
extern EthernetServer server;
extern char ipchar[16];
extern char subnetchar[16];
extern IPAddress subnet;
extern char ipadress;
extern char subnetadress;
extern String ipip;
extern String subnetsubnet;
extern String ethernetInfo[5];


// Icons for Menus
extern const unsigned short *images_main[];
extern const unsigned short *images_settings[];
extern const unsigned short *images_stepper[];

// Screen Control Variables
extern bool screenOn;
extern bool engineeringMode;
extern volatile int encoderState;
extern RtcDateTime lastDisplayedTime;

// Queues and Synchronization
extern QueueHandle_t screenUpdateQueue;
extern xSemaphoreHandle gatekeeper;

enum ScreenUpdateCommand {
  UPDATE_MAIN_MENU,
  UPDATE_CONFIG_BRIGHTNESS,
  UPDATE_CONFIG_BACKGROUNDCOLOR,
  UPDATE_CONFIG_ASPECT
};

// Color Arrays for Background, Selection Bar, Border, and Text
extern const uint16_t backgroundpossibleColors[];
extern const uint16_t selectbarpossibleColors[];
extern const uint16_t borderpossibleColors[];
extern const uint16_t textpossibleColors[];

// Color Count Variables
extern const int backgroundcolors;
extern const int selectbarcolors;
extern const int textcolors;
extern const int bordercolors;
extern const int selectnumcolors[];

// Color Menu Arrays
extern MenuState colormenuArray[];
extern String background[];
extern String selectionbar[];
extern String border[];
extern String text[];

// Selected Colors
extern int selectedBackgroundColor;
extern int selectedTextColor;
extern int selectedBarColor;
extern int selectedBorderColor;
extern int *selection[];

// Font Configuration
extern const GFXfont* possibleFont12[];
extern const GFXfont* possibleFont9[];
extern const int numFonts;
extern String fontType[];
extern const GFXfont* textType12;
extern const GFXfont* textType9;

extern BfButton btn;
extern ThreeWire myWire;
extern RtcDS1302<ThreeWire> Rtc;
extern AccelStepper motor;
extern TFT_eSPI tft;
extern TickType_t previousTick;
extern TickType_t previousTime;


#endif // GLOBALS_H
```

3.24 GUI/GUI.cpp File Reference

Implementation of the functions related to the graphical user interface (GUI).

```
#include "GUI.h"
```

Functions

- void printDateTime (const RtcDateTime &dt)

  Displays the date and time on the screen and in the serial monitor.
- void drawNeedle (int angle, int length, uint32_t color)

  Draws a needle on the dial.

- void drawDialMarks ()

    Draws the dial marks.
- void updateDial ()

    Updates the dial by moving the needle to the new position.
- void setBrightness (int level)

    Sets the screen brightness level.
- void drawBrightnessBar (int level)

    Draws the brightness bar on the screen.
- void drawSelectionBar (int16_t y, bool isSelected, int currentIndex, int totalItems)

    Draws a selection bar on the screen.

3.24.1 Detailed Description

Implementation of the functions related to the graphical user interface (GUI).

This file contains the implementation of the functions to handle the GUI, such as drawing the dial needle, selection bar, and other functionalities.

Author

Juan Alberto Serrano Redondo.

3.24.2 Function Documentation

3.24.2.1 drawBrightnessBar()

```
void drawBrightnessBar (
             int level)
```

Draws the brightness bar on the screen.

Parameters

level	Current brightness level.

3.24.2.2 drawDialMarks()

```
void drawDialMarks ()
```

Draws the dial marks.

Draws the marks and numbers at regular intervals around the dial.

3.24.2.3 drawNeedle()

```
void drawNeedle (
             int angle,
             int length,
             uint32_t color)
```

Draws a needle on the dial.

This function draws a needle on the dial based on the specified angle and length.

Parameters

angle	Angle of the needle.
length	Length of the needle.
color	Color of the needle.

3.24.2.4 drawSelectionBar()

```
void drawSelectionBar (
            int16_t y,
            bool isSelected,
            int currentIndex,
            int totalItems)
```

Draws a selection bar on the screen.

This function draws a selection bar at the specified vertical position. The bar will be drawn differently based on whether it is selected or not, and its appearance can change depending on the current and total items.

Parameters

y	Vertical position of the bar (in pixels).
isSelected	A boolean flag indicating whether the bar is selected.
currentIndex	Index of the currently selected item (starting from 0).
totalItems	Total number of items in the selection list.

3.24.2.5 printDateTime()

```
void printDateTime (
            const RtcDateTime & dt)
```

Displays the date and time on the screen and in the serial monitor.

This function takes a RtcDateTime object, formats the time as HH:MM, and displays it both on the serial monitor and the TFT screen.

Parameters

dt	The current date and time in RtcDateTime format.

3.24.2.6 setBrightness()

```
void setBrightness (
            int level)
```

Sets the screen brightness level.

Parameters

level	The brightness level to set.

3.24.2.7 updateDial()

```
void updateDial ()
```

Updates the dial by moving the needle to the new position.

This function is responsible for refreshing the dial display by updating the needle's position based on new data or changes in the system state.

3.25 GUI/GUI.h File Reference

Graphical User Interface (GUI) module for managing the display elements.

```
#include <globals/globals.h>
#include <libraries.h>
```

Functions

- void drawNeedle (int angle, int length, uint32_t color)

  Draws a needle on the dial.
- void drawDialMarks ()

  Draws the dial marks.
- void drawSelectionBar (int16_t y, bool isSelected, int currentIndex, int totalItems)

  Draws a selection bar on the screen.
- void updateDial ()

  Updates the dial by moving the needle to the new position.
- void setBrightness (int level)

  Sets the screen brightness level.
- void drawBrightnessBar (int level)

  Draws the brightness bar on the screen.
- void printDateTime (const RtcDateTime &dt)

  Displays the date and time on the screen and in the serial monitor.

3.25.1 Detailed Description

Graphical User Interface (GUI) module for managing the display elements.

This header file contains the function declarations for managing the graphical user interface (GUI) of the system. The functions provided here allow drawing different UI elements like needles, dials, selection bars, and adjusting the screen's brightness. This module interacts with the hardware display and manages visual elements based on user input or system state changes.

The primary purpose of this module is to offer functions that facilitate displaying information in a structured and visually intuitive manner on the screen, including dynamic updates such as moving the needle on a dial, changing brightness levels, and displaying the date and time.

It is assumed that the GUI interacts with a TFT display and uses a rotary encoder for user input, as well as other UI elements such as selection bars for menu navigation.

The file also defines various graphical components like brightness bars, dials, and selection bars.

Dependencies:

- This file relies on the global variables defined in `globals.h` and the libraries included in `libraries.h`.

Author

Juan Alberto Serrano Redondo.

3.25.2 Function Documentation

3.25.2.1 drawBrightnessBar()

```
void drawBrightnessBar (
             int level)
```

Draws the brightness bar on the screen.

This function draws a visual representation of the current brightness level on the screen, showing the user the current setting.

Parameters

level	Current brightness level.
level	Current brightness level.

3.25.2.2 drawDialMarks()

```
void drawDialMarks ()
```

Draws the dial marks.

This function draws the marks around the dial, including numbers at angles 0, 60, 120, etc. It is used to set up the dial's appearance with evenly spaced markings.

Draws the marks and numbers at regular intervals around the dial.

3.25.2.3 drawNeedle()

```
void drawNeedle (
             int angle,
             int length,
             uint32_t color)
```

Draws a needle on the dial.

This function draws a needle on the dial at the specified angle and length, using the given color.

Parameters

angle	The angle at which to draw the needle (in degrees).
length	The length of the needle.
color	The color of the needle, represented as a 32-bit unsigned integer (usually RGB color).

This function draws a needle on the dial based on the specified angle and length.

Parameters

angle	Angle of the needle.
length	Length of the needle.
color	Color of the needle.

3.25.2.4 drawSelectionBar()

```
void drawSelectionBar (
             int16_t y,
             bool isSelected,
             int currentIndex,
             int totalItems)
```

Draws a selection bar on the screen.

This function draws a selection bar at the specified vertical position. The bar will be drawn differently based on whether it is selected or not, and its appearance can change depending on the current and total items.

Parameters

y	Vertical position of the bar (in pixels).
isSelected	A boolean flag indicating whether the bar is selected.
currentIndex	Index of the currently selected item (starting from 0).
totalItems	Total number of items in the selection list.

3.25.2.5 printDateTime()

```
void printDateTime (
             const RtcDateTime & dt)
```

Displays the date and time on the screen and in the serial monitor.

This function prints the current date and time on the screen, as well as in the serial monitor for debugging purposes.

Parameters

dt	The current date and time in RtcDateTime format, typically retrieved from a real-time clock (RTC).

This function takes a RtcDateTime object, formats the time as HH:MM, and displays it both on the serial monitor and the TFT screen.

Parameters

dt	The current date and time in RtcDateTime format.

3.25.2.6 setBrightness()

```
void setBrightness (
            int level)
```

Sets the screen brightness level.

This function adjusts the brightness level of the screen based on the provided level value.

Parameters

level	The brightness level to set (usually between a predefined min and max value).
level	The brightness level to set.

3.25.2.7 updateDial()

```
void updateDial ()
```

Updates the dial by moving the needle to the new position.

This function is responsible for refreshing the dial display by updating the needle's position based on new data or changes in the system state.

3.26 GUI.h

Go to the documentation of this file.

```
#ifndef GUI_H
#define GUI_H

#include <globals/globals.h>
#include <libraries.h>

void drawNeedle(int angle, int length, uint32_t color);

void drawDialMarks();

void drawSelectionBar(int16_t y, bool isSelected, int currentIndex, int totalItems);

void updateDial();

void setBrightness(int level);

void drawBrightnessBar(int level);

void printDateTime(const RtcDateTime& dt);

#endif // GUI_H
```

3.27 automatic_icon.h

```
#if defined(__AVR__)
    #include <avr/pgmspace.h>
#elif defined(__PIC32MX__)
    #define PROGMEM
#elif defined(__arm__)
    #define PROGMEM
#endif

const unsigned short automatic_icon[ ] PROGMEM={0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0020, 0x0020, 0x0020, 0x0020, 0x0020, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0020, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0020, 0x0000, 0x2945, 0x2965, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x4a69, 0x8430, 0x7bcf, 0x0020, 0x0000, 0x0000, 0x5aeb, 0xffff, 0xffff, 0x7bcf,
0x2104, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x0000, 0x0020, 0x0020, 0x0000,
0x0000, 0x5acb, 0xffff, 0x52aa, 0xb5b6, 0xc638, 0xad55, 0xad55, 0xffff, 0xad55, 0xa514, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x18e3,
0x5acb, 0xffff, 0xf7be, 0x10a2, 0x8430, 0xffff, 0xffff, 0xffff, 0xf79e, 0x10a2, 0x0861, 0x39e7,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6b6d, 0x73ae, 0xe73c,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xa534, 0x8430, 0x8410, 0xffdf, 0xd69a, 0xc638, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x8430, 0xffff, 0xffff, 0xffff,
0xffff, 0xffdf, 0x738e, 0x39e7, 0x4228, 0x0000, 0x0000, 0x0000, 0x39e7, 0xffff, 0xffff, 0x52aa,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x2104, 0x0000, 0x0020, 0x0000, 0x632c, 0xffff, 0x0000, 0x630c, 0xffff,
0x9cf3, 0x0861, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0020, 0x0000, 0x0841, 0x0841, 0x0000,
0x0000, 0x0000, 0x2104, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x73ae, 0xffff, 0x73ae, 0xb5b6, 0xffdf,
0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x18e3, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x18c3, 0xc618, 0xffff, 0xffff, 0xffdf,
0xc638, 0x632c, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x2104, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0020, 0x5acb, 0xffff,
0xffff, 0xffff, 0xffff, 0x2965, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4a49,
0xffff, 0xbdd7, 0x7bcf, 0xef7d, 0x39c7, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x7bcf,
0xffff, 0x7bcf, 0x0000, 0xffff, 0xef7d, 0x18e3, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1082, 0xdedb,
0xffdf, 0xffdf, 0xffff, 0xffdf, 0xffff, 0x6b4d, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x0000, 0x18c3, 0xce79, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x6b4d, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0841, 0x0000, 0x3186, 0xffdf, 0xe73c,
0xef7d, 0xef7d, 0xef7d, 0xef7d, 0xef5d, 0xef5d, 0xbdf7, 0x0000, 0x0020, 0x0000, 0x0020, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0020, 0x0020, 0x0020, 0x0020, 0x0020, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff};
```

3.28 brightness_icon.h

```
#if defined(__AVR__)
    #include <avr/pgmspace.h>
#elif defined(__PIC32MX__)
    #define PROGMEM
#elif defined(__arm__)
    #define PROGMEM
#endif

const unsigned short brightness_icon[ ] PROGMEM={0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff};
```

3.29 color_icon.h

```
#if defined(__AVR__)
    #include <avr/pgmspace.h>
#elif defined(__PIC32MX__)
    #define PROGMEM
#elif defined(__arm__)
    #define PROGMEM
#endif

const unsigned short color_icon[ ] PROGMEM={0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffdf, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffdf, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffdf, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xf7be, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffdf, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffdf, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffdf, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffdf, 0xffff, 0xffff,
0xffff, 0xffdf, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xffdf, 0xffff,
0xffdf, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffdf,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0x0000, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0x0000, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffdf,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffdf, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xf7be, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xf7be, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffdf, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffdf, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffdf, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffdf,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffdf, 0xffdf, 0xffff, 0xffff, 0xf7be,
0xef7d, 0xef7d, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffdf, 0xffdf, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff};
```

3.30 empty_sun.h

```
// Generated by   : ImageConverter 565 Online
// Generated from : luna.jpg
// Time generated : Sun, 07 Jul 24 20:42:29 +0200  (Server timezone: CET)
// Image Size     : 50x49 pixels
// Memory usage   : 4900 bytes


#if defined(__AVR__)
    #include <avr/pgmspace.h>
#elif defined(__PIC32MX__)
    #define PROGMEM
#elif defined(__arm__)
    #define PROGMEM
#endif

const unsigned short empty_sun[2450] PROGMEM={
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x4a49, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff};
```

3.31 ethernet_active.h

```
// Generated by   : ImageConverter 565 Online
// Generated from : luna.jpg
// Time generated : Sun, 07 Jul 24 20:42:29 +0200  (Server timezone: CET)
// Image Size      : 50x49 pixels
// Memory usage    : 4900 bytes


#if defined(__AVR__)
    #include <avr/pgmspace.h>
#elif defined(__PIC32MX__)
    #define PROGMEM
#elif defined(__arm__)
    #define PROGMEM
#endif

const unsigned short ethernet_active[2450] PROGMEM={0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0x0000,
0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0x0000,
0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0x0000,
0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0x0000,
0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0x0000,
0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000};
```

3.32 ethernet_icon.h

```
// Generated by   : ImageConverter 565 Online
// Generated from : luna.jpg
// Time generated : Sun, 07 Jul 24 20:42:29 +0200  (Server timezone: CET)
// Image Size      : 50x49 pixels
// Memory usage    : 4900 bytes


#if defined(__AVR__)
    #include <avr/pgmspace.h>
#elif defined(__PIC32MX__)
    #define PROGMEM
#elif defined(__arm__)
    #define PROGMEM
#endif

const unsigned short ethernet_icon[] PROGMEM={0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0x0000, 0xffff, 0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0x0000, 0xffff, 0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0x0000, 0xffff, 0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0x0000, 0xffff, 0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff};
```

3.33 font_icon.h

```
#if defined(__AVR__)
    #include <avr/pgmspace.h>
#elif defined(__PIC32MX__)
    #define PROGMEM
#elif defined(__arm__)
    #define PROGMEM
#endif

const unsigned short font_icon[ ] PROGMEM={0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0xffff, 0x0000, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000,
0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0x0000, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff};
```

3.34 full_sun.h

```
// Generated by   : ImageConverter 565 Online
// Generated from : sol.jpg
// Time generated : Sun, 07 Jul 24 20:42:51 +0200  (Server timezone: CET)
// Image Size     : 50x49 pixels
// Memory usage   : 4900 bytes


#if defined(__AVR__)
    #include <avr/pgmspace.h>
#elif defined(__PIC32MX__)
    #define PROGMEM
#elif defined(__arm__)
    #define PROGMEM
#endif

const unsigned short full_sun[2450] PROGMEM={0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff};

```

3.35 granasat_logo.h

```
// Generated by   : ImageConverter 565 Online
// Generated from : images.png
// Time generated : Wed, 17 Jul 24 21:08:31 +0200  (Server timezone: CET)
// Image Size     : 225x225 pixels
// Memory usage   : 101250 bytes


#if defined(__AVR__)
    #include <avr/pgmspace.h>
#elif defined(__PIC32MX__)
    #define PROGMEM
#elif defined(__arm__)
    #define PROGMEM
#endif

const unsigned short granasat_logo[] PROGMEM={0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffdf, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xd73d, 0x9e3b, 0x85da, 0x6d79, 0x6538,
0x54d8, 0x4497, 0x3476, 0x3456, 0x2c36, 0x2c36, 0x2c36, 0x3456, 0x4497, 0x54d8, 0x6538, 0x6d79,
0x85da, 0xa65b, 0xdf3d, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffdf, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xef9e, 0xc6fd, 0xa63b,
0x6d59, 0x44d8, 0x3c77, 0x2c36, 0x2436, 0x2436, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x2c56, 0x3456, 0x3456, 0x2c36, 0x2c36, 0x2436, 0x2436, 0x2436, 0x2c56, 0x4497, 0x4cd8, 0x6d59,
0xa63b, 0xcf1d, 0xef9e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xdf5e, 0x9e3b, 0x5d38, 0x3c77, 0x1c16, 0x1bf5,
0x0bb5, 0x2c56, 0x2c56, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0x1bf5, 0x2c36, 0x2c36, 0x2c36,
0x2416, 0x13d5, 0x1bf5, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0x2416, 0x1bf5,
0x1bf6, 0x1c15, 0x3c97, 0x5d38, 0x9e3b, 0xdf5e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xdf5e, 0x8dfa, 0x54f8, 0x2416, 0x1bf5, 0x1bf5, 0x2c36, 0x2c36, 0x3456,
0xb69c, 0xd73d, 0xa65b, 0x2416, 0x2c36, 0x2436, 0x3c77, 0x5d18, 0x7dba, 0x2c56, 0x2c36, 0x2416,
0x54d8, 0xa67c, 0x7dba, 0x2436, 0x2c36, 0x2c36, 0x2c36, 0x3476, 0x2c36, 0x2416, 0x2c36, 0x3456,
0x3456, 0x3456, 0x2c36, 0x2416, 0x1bf5, 0x2416, 0x54f8, 0x961a, 0xdf5e, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xf7df, 0xae7c, 0x5d18, 0x2416, 0x1bf5, 0x2416, 0x2c36, 0x3456, 0x2c56, 0x3456, 0x1bf5, 0xb69c,
0xb69c, 0x13f5, 0x2456, 0x2c36, 0x2c36, 0x1bf6, 0x6538, 0xffdf, 0x7dda, 0x2c36, 0x2416, 0x3c77,
0xdf3d, 0x7579, 0xc6fd, 0x85ba, 0x1bf5, 0x2416, 0x4cd8, 0xcefd, 0x2c36, 0x961a, 0x4cd7, 0x2416,
0x2c36, 0x2c36, 0x1c16, 0x1bf5, 0x2c36, 0x2c36, 0x2416, 0x1bf5, 0x2416, 0x5d18, 0xae7c, 0xf7df,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xe77e, 0x8dfa,
0x3c97, 0x1bf5, 0x1bf6, 0x2c36, 0x2c56, 0x2c56, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0xe77e,
0x3477, 0x7579, 0xef9e, 0x6d59, 0x2416, 0x2416, 0x54d8, 0xd71d, 0x13d5, 0x2c36, 0x2416, 0x54f8,
0xdf5e, 0x0354, 0x965b, 0xa65b, 0x1bd5, 0x1bf5, 0x6d79, 0xbedd, 0x13d5, 0xdf3d, 0x4cd7, 0x2416,
0x2416, 0x5d18, 0xae5b, 0x7599, 0x3c97, 0x2416, 0x3456, 0x2c36, 0x2c36, 0x1bf5, 0x1bf5, 0x3c97,
0x8dfa, 0xe77e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xe77e, 0x7dba, 0x2c56, 0x1bf5,
0x2416, 0x3456, 0x3456, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x1bf6, 0xcefd,
0xa65b, 0x0bb5, 0x9e3b, 0xbebc, 0x13d5, 0x2c36, 0x3477, 0xdf7e, 0x2c56, 0x2c36, 0x2c36, 0x2416,
0xa65b, 0xcf1d, 0xcefd, 0x44b7, 0x2c36, 0x1bf5, 0x759a, 0xd75e, 0x6579, 0xe77e, 0x2436, 0x2c56,
0x1bd5, 0x9e3b, 0xc6fd, 0x961b, 0xc6fd, 0x2416, 0x2c36, 0x2c36, 0x2c36, 0x3456, 0x3456, 0x2416,
0x1bf5, 0x2c56, 0x7dba, 0xdf5e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xef9e, 0x85ba, 0x2c56, 0x1bf5, 0x2c36, 0x3456,
0x2c56, 0x2c56, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0x4cd7,
0xdf5e, 0xcf1d, 0xc6fd, 0x7d99, 0x1c16, 0x2c36, 0x2c36, 0x44b7, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x2416, 0x44d7, 0x3456, 0x2416, 0x2c36, 0x2c36, 0x3456, 0x961b, 0xa65b, 0x9e5b, 0x1c16, 0x3436,
0x1c16, 0xdf5e, 0x44b7, 0x3497, 0xdf5e, 0x2416, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x3456, 0x2c36, 0x1bf5, 0x2c56, 0x85da, 0xef9e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xa67b, 0x3477, 0x1bf5, 0x2c36, 0x3456, 0x2416, 0x2436,
0x2416, 0x2416, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2416,
0x2c56, 0x4cf8, 0x3477, 0x1bf5, 0x2c36, 0x2c36, 0x2c36, 0x2415, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x2c36, 0x2415, 0x2c16, 0x2c36, 0x2c36, 0x2c36, 0x2c16, 0x13f5, 0x1bf5, 0x2416, 0x2c36, 0x2416,
0x44b7, 0xe75e, 0xbebc, 0xe79e, 0x7d9a, 0x1bf5, 0x3456, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x2c36, 0x2c36, 0x3456, 0x2c36, 0x1bf5, 0x3477, 0xae7c, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xd73d, 0x54f8, 0x1bf5, 0x2436, 0x3456, 0x2c36, 0x2416, 0x3c97, 0xa65b,
0xbebc, 0x54f8, 0x2416, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x2c16, 0x23f5, 0x2c16, 0x3456, 0x2c57, 0x2c77, 0x2c77, 0x3498, 0x3498, 0x3498, 0x2c98, 0x2c78,
0x2c98, 0x3498, 0x3498, 0x2c98, 0x2c78, 0x2c77, 0x2c77, 0x3477, 0x3456, 0x2c36, 0x3436, 0x13d5,
0x8dda, 0xae7c, 0x3477, 0x5518, 0x2416, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x2c36, 0x2c36, 0x2c36, 0x1bf5, 0x2416, 0x13d5, 0x0bb5, 0x54f8, 0xd73d, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x961b, 0x2436, 0x1bf6, 0x3456, 0x2c36, 0x2c16, 0x3456, 0x1bf5, 0xae7b, 0x95fb,
0xc6dd, 0x7599, 0x1bf5, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c57,
0x3477, 0x3498, 0x2c77, 0x2c15, 0x2393, 0x2351, 0x1acf, 0x1a4c, 0x11ea, 0x1188, 0x0947, 0x0947,
0x0947, 0x0947, 0x1188, 0x120a, 0x1a6d, 0x1acf, 0x2351, 0x2393, 0x2c15, 0x2c77, 0x3498, 0x2c57,
0x44b8, 0x44b7, 0x1bf5, 0x2415, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x2c36, 0x2c36, 0x2416, 0x7dba, 0xc6fd, 0xc6dc, 0x5d38, 0x13d5, 0x2436, 0x95fa, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xdf5e, 0x4cf8, 0x13d5, 0x2c36, 0x3456, 0x2416, 0x3456, 0x3477, 0x2c36, 0x1bf5, 0x8dda, 0xf7bf,
0x5d38, 0x8dfa, 0x2c56, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c57, 0x2c98, 0x2c56, 0x23b3,
0x1aef, 0x11a9, 0x0926, 0x0083, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x00a3, 0x0946, 0x11c9, 0x22ef,
0x2393, 0x2436, 0x3498, 0x2c77, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x3436, 0x1bf5, 0x7d99, 0xefbe, 0x4cd8, 0x5518, 0xcf1d, 0x4cd7, 0x2416, 0x1bd5, 0x4cf8, 0xdf5e,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xb6bc,
0x2c36, 0x1c16, 0x3456, 0x2c36, 0x2416, 0x44b7, 0xcf1d, 0x5d18, 0x2416, 0x2c36, 0x2436, 0x9e3b,
0xb6bc, 0x6d59, 0x2416, 0x2c36, 0x2c36, 0x2c57, 0x3498, 0x2c36, 0x2372, 0x120a, 0x0905, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0926, 0x122b, 0x2372, 0x2c36, 0x3498, 0x2c57, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x2c36, 0x1bf5, 0xdf3e, 0x5518, 0x4cd8, 0x4cf8, 0x2c56, 0x3476, 0x2c36, 0x3456, 0x1c16, 0x2c36,
0xb6bc, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x8dda, 0x13f5,
0x2c36, 0x3456, 0x2c36, 0x3456, 0x13d5, 0xb69c, 0x7dba, 0x0374, 0x2c36, 0x2c36, 0x2c36, 0x1c16,
0x2416, 0x1bf5, 0x2c36, 0x2c77, 0x2c77, 0x23b3, 0x122b, 0x08c4, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x08c4, 0x122b, 0x23b3, 0x2c77, 0x2c77, 0x2c36, 0x2c36, 0x2c36,
0x3456, 0x1bf5, 0xbedc, 0x7dda, 0x44b7, 0xffff, 0x54d8, 0x1bf6, 0x3456, 0x2c36, 0x3456, 0x2c36,
0x13f5, 0x85da, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xf7df, 0x6d59, 0x13d5, 0x2c36,
0x2c56, 0x2416, 0x2c36, 0x3456, 0x1bf5, 0x8dfa, 0xcf1d, 0x6d99, 0x9e3b, 0x2416, 0x2c36, 0x2c36,
0x2c36, 0x3477, 0x2c57, 0x2330, 0x0967, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0967, 0x2310, 0x2c57, 0x2c77, 0x2c36,
0x2c36, 0x2436, 0x3c97, 0xc6fd, 0xcf1d, 0xbebc, 0x2416, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0x2436,
0x3456, 0x13d5, 0x6d59, 0xf7df, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xef9e, 0x4cf8, 0x1bf5, 0x3456, 0x2c36,
0x2416, 0x5518, 0x3456, 0x2416, 0x2c36, 0x2436, 0x9e3b, 0xbebc, 0x54d8, 0x2416, 0x2c36, 0x2c77,
0x2c57, 0x2310, 0x0906, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0905, 0x2310, 0x2c57,
0x2c77, 0x2c36, 0x2416, 0x2436, 0x6d59, 0x4497, 0x2416, 0x2c36, 0x1bf6, 0xcf1d, 0xa65b, 0x3456,
0x2c36, 0x3456, 0x1bf5, 0x4cf8, 0xef9e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xe77e, 0x3c97, 0x1bf5, 0x3456, 0x2c36, 0x2416,
0x85da, 0xc6fd, 0xdf5e, 0x3456, 0x2416, 0x2c36, 0x1bf5, 0x13d5, 0x23f5, 0x2c57, 0x2c77, 0x2331,
0x0926, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0926,
0x2331, 0x2c77, 0x2c57, 0x2c36, 0x1bf6, 0x2c36, 0x3456, 0x1bf5, 0xae7c, 0xcf1d, 0x961a, 0x5d18,
0x2416, 0x3456, 0x3456, 0x1bf5, 0x3c97, 0xe77e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xdf5e, 0x3c97, 0x1c16, 0x3456, 0x2c36, 0x2c36, 0x2c36,
0x759a, 0x6d79, 0xbebc, 0xcefd, 0x2c36, 0x2c36, 0x3436, 0x3436, 0x3498, 0x2bf5, 0x11c9, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x11a9, 0x2bd4, 0x2c78, 0x2c36, 0x2c36, 0x2416, 0x54f8, 0xd71d, 0x2436, 0x13f5, 0x2c36,
0x3456, 0x2416, 0x2416, 0x3456, 0x1c16, 0x3c97, 0xdf5e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xe75e, 0x3c77, 0x1c16, 0x3456, 0x2c36, 0x2c36, 0x3456, 0x2436,
0x2c56, 0xcf1d, 0x961a, 0xa65b, 0x2c56, 0x2c16, 0x2c57, 0x2c77, 0x1aae, 0x0062, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0062, 0x1a8d, 0x2c77, 0x2c56, 0x2c36, 0x2c56, 0x2c36, 0x2436, 0x3456, 0x2c36,
0x13d5, 0x54f8, 0x54f8, 0x1bf5, 0x3456, 0x1c16, 0x3c97, 0xe77e, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xef9e, 0x3c97, 0x2416, 0x3456, 0x2c36, 0x2c36, 0x2c56, 0x1bf6, 0x2c56,
0x2416, 0x85da, 0x9e1b, 0x1bf5, 0x2c15, 0x3478, 0x2bf5, 0x0967, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2124, 0x10a2, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0967, 0x2bf5, 0x2c77, 0x2c16, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x7d99, 0x7d99, 0xb67c, 0x5d18, 0x2416, 0x3456, 0x1c16, 0x44b7, 0xef9e, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xf7df, 0x4cd8, 0x1bf6, 0x3456, 0x2c36, 0x13f5, 0xae7c, 0xe75e, 0x6d79, 0x1bf5,
0x2c56, 0x1bf5, 0x1bf5, 0x2c36, 0x3498, 0x2351, 0x0083, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4208, 0x2945, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0083, 0x2351, 0x2c98, 0x2c36, 0x2c36, 0x1bf5, 0x7d99,
0xd73d, 0x7d9a, 0xd75e, 0x7dba, 0x1bf5, 0x2c56, 0x3456, 0x1bf6, 0x4cd8, 0xf7df, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x6d59, 0x1bf5, 0x3456, 0x2c36, 0x2436, 0xa65b, 0x8dfa, 0x3c97, 0xef9e, 0x3c97,
0x2416, 0x2c36, 0x2c36, 0x2c78, 0x1a6d, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x1082, 0x18c3, 0x6b6d, 0x4a49, 0x0861, 0x0020, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1a6d, 0x2c78, 0x2c36, 0x2416, 0x44b7,
0x9e3b, 0xcf1d, 0xb69c, 0x1bf6, 0x2c36, 0x2c36, 0x2c36, 0x3456, 0x1bf5, 0x6d59, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0x8dda, 0x13d5, 0x3456, 0x2c36, 0x2c36, 0x2436, 0xc6dd, 0xcefd, 0x2416, 0xd71d, 0x54d8,
0x2415, 0x2c56, 0x2c57, 0x11ea, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0861, 0x2965, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0861, 0x39e7, 0x632c, 0xad55, 0xce79, 0x9cd3, 0x6b6d, 0x10a2, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x11ea, 0x2c57, 0x2c56, 0x23f5,
0x3477, 0x961b, 0x1c16, 0x2c36, 0x2c56, 0x1bf5, 0x2416, 0x2c36, 0x3456, 0x13d5, 0x8dfa, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xb6bc, 0x13f5, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x1bf5, 0xa65b, 0xffff, 0x7579, 0x1bf5,
0x3457, 0x2c56, 0x1188, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0020, 0x4208, 0x8430, 0x18e3, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4228, 0x8430, 0x10a2, 0x2965, 0x528a, 0x4a49,
0x39e7, 0x1082, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1188, 0x2c56, 0x2c56,
0x2c16, 0x1bf5, 0x2c56, 0x2c36, 0x13d5, 0x8e1a, 0xbebc, 0x13d5, 0x2c36, 0x2c56, 0x13f5, 0xb6bc,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xdf5e,
0x2c36, 0x2c36, 0x2c56, 0x2416, 0x3c77, 0x2c56, 0x2c36, 0x2c36, 0x13d5, 0x9e3b, 0xcf1d, 0x2c57,
0x2c36, 0x0946, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x2124, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0841, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x31a6, 0x4228, 0x4228, 0x0000, 0x0000, 0x10a2,
0x2945, 0x528a, 0x4a69, 0x39e7, 0x2104, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0946, 0x2c36,
0x2c56, 0x3436, 0x2c36, 0x3456, 0xb69c, 0xd73d, 0x961b, 0x9e3b, 0x2416, 0x2c56, 0x2c36, 0x2c36,
0xdf5e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x4cf8,
0x1c16, 0x3456, 0x2416, 0x44b7, 0xae5b, 0x3477, 0x3456, 0x2c36, 0x2c56, 0x1bf5, 0x6d79, 0x3477,
0x0926, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1082, 0x6b4d, 0x3186, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x18c3, 0x0841, 0x4a49, 0x2104, 0x0000, 0x0000,
0x0000, 0x0000, 0x0020, 0x2965, 0x4208, 0x52aa, 0x4228, 0x2965, 0x0020, 0x0000, 0x0000, 0x0000,
0x0000, 0x0020, 0x31a6, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0946,
0x2c57, 0x2c56, 0x2416, 0x54d8, 0xae7c, 0x03b5, 0x6d79, 0xef9e, 0x2436, 0x2c36, 0x3456, 0x1bf6,
0x54f8, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x95fa, 0x13d5,
0x3456, 0x3456, 0x1bf5, 0x85ba, 0xd71d, 0xb69c, 0xc6fd, 0x3c77, 0x2c36, 0x2c36, 0x2436, 0x0967,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1082, 0x632c, 0x3186, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1082, 0x5acb, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2124, 0x39e7, 0x52aa, 0x4a49, 0x2965, 0x1082,
0x0000, 0x0000, 0x632c, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0968, 0x2c77, 0x2c36, 0x2c36, 0x0bb5, 0x961b, 0xef9e, 0x44b7, 0x2416, 0x2c36, 0x2c36, 0x3456,
0x13d5, 0x961b, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xd73d, 0x2436, 0x2c36,
0x2c36, 0x2c36, 0x2436, 0x3c97, 0x961a, 0x6d59, 0xc6dc, 0x3c97, 0x2416, 0x3498, 0x11ea, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x39e7, 0x3186, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1082, 0x39c7, 0x5acb,
0x632c, 0x4a49, 0x9cd3, 0x2945, 0x2104, 0x1082, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x11ea, 0x2c98, 0x2c36, 0x2c36, 0x85ba, 0x3c97, 0x1bf6, 0x3456, 0x2c36, 0x2416, 0x2c36,
0x2c36, 0x2436, 0xd73d, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x54f8, 0x2416, 0x2c56,
0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x0bb5, 0x6d59, 0x6d59, 0x1bd5, 0x3498, 0x1a6d, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0841, 0x5acb, 0x0020,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x18c3,
0x7bef, 0xc638, 0xce79, 0x7bef, 0x528a, 0x18e3, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x1a6d, 0x2c98, 0x2c16, 0x1bf5, 0x2416, 0x2c36, 0x13d5, 0x2c56, 0x54f8, 0x1c16,
0x3456, 0x2416, 0x54f8, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xae7c, 0x1bf5, 0x3456, 0x2c36,
0x2416, 0x2436, 0x1c16, 0x2416, 0x3456, 0x2416, 0x23f5, 0x3478, 0x2351, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3186, 0x39e7,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0861, 0x4a49,
0x4a69, 0x10a2, 0x73ae, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x2351, 0x2c77, 0x2c36, 0x2c36, 0x3456, 0x961b, 0x7599, 0xae7b, 0x54f8,
0x2416, 0x3456, 0x1bf5, 0xae7c, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xefbf, 0x3477, 0x2416, 0x3456, 0x2416,
0x5518, 0xd73d, 0xc6dd, 0x4497, 0x2416, 0x2c36, 0x2c57, 0x2bf5, 0x0082, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x08c4, 0x0968, 0x08e4, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x52aa,
0x0841, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3186, 0x5aeb, 0x31a6,
0x0000, 0x0000, 0x528a, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0082, 0x2bf5, 0x3457, 0x1bf5, 0x7d99, 0xd71d, 0x85da, 0xcf1d, 0x8e1b,
0x13d5, 0x3456, 0x2416, 0x3477, 0xefbf, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x85da, 0x1bf5, 0x3456, 0x2c36, 0x2c56,
0xcefd, 0x3477, 0x95fb, 0xbebc, 0x1bf5, 0x2c36, 0x2c77, 0x0946, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0062, 0x1ace, 0x1a8e, 0x00a3, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x18c3,
0x4a69, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2124, 0x5aeb, 0x4208, 0x0020, 0x0000,
0x0000, 0x0000, 0x18c3, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020,
0x0000, 0x0000, 0x0000, 0x0000, 0x0967, 0x2c77, 0x2c36, 0x2c56, 0x9e3b, 0xd73d, 0xbedc, 0x3456,
0x2c36, 0x2c36, 0x3456, 0x13f5, 0x85da, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xe77e, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c56,
0xdf5e, 0x5d18, 0x6d59, 0xb69c, 0x1bf5, 0x3498, 0x1a8d, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x08c4,
0x2c15, 0x1a6d, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1082, 0x2104, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x4a69, 0x18e3, 0x0000, 0x0000, 0x0000, 0x0861, 0x4a49, 0x528a, 0x1082, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0841, 0x5aeb,
0x2945, 0x0000, 0x0000, 0x0000, 0x0000, 0x1aae, 0x3498, 0x1bf5, 0x54f8, 0x7599, 0x1bf5, 0x2416,
0x2c36, 0x2416, 0x2416, 0x2c36, 0x2c36, 0xe77e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x85ba, 0x1bf5, 0x3456, 0x2c36, 0x3456, 0x2416,
0x5d18, 0xd71d, 0xcefd, 0x44b7, 0x2436, 0x2bf4, 0x0041, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x00c4, 0x2c57,
0x1ace, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3186, 0x4228, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x1082, 0x52aa, 0x0000, 0x0000, 0x39c7, 0x5acb, 0x2945, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x10a2, 0x738e,
0x31a6, 0x0000, 0x0000, 0x0000, 0x0000, 0x0062, 0x2bf4, 0x2c57, 0x2416, 0x1bf6, 0x2c36, 0x2c36,
0x2416, 0x44b7, 0x54f8, 0x2416, 0x1bf5, 0x7dba, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xe77e, 0x2c56, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0x2c36,
0x2416, 0x2416, 0x2416, 0x2415, 0x3478, 0x1188, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x2372, 0x2c15,
0x0021, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x528a, 0x528a, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x4a69, 0x6b6d, 0x528a, 0x39c7, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1082,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x11a8, 0x2c77, 0x2c36, 0x3456, 0x2c36, 0x2c36,
0x3c97, 0xdf3d, 0xdf5e, 0x6d79, 0x1bf5, 0x3456, 0xe77e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x8dfa, 0x13f5, 0x3456, 0x2c36, 0x3456, 0x6538, 0x2c36,
0x3456, 0x2c36, 0x2c36, 0x3477, 0x2331, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0967, 0x34d9, 0x120a,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0841, 0x4228, 0x6b6d, 0xb5b6, 0xb5b6, 0x6b6d, 0x4228, 0x1082, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x3186, 0x9cf3, 0x2104, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2331, 0x3477, 0x13d5, 0x85da, 0x6d79,
0x44d8, 0xcefd, 0x3c77, 0xd73d, 0x3456, 0x13d5, 0x8dfa, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xf7df, 0x3c77, 0x2416, 0x3456, 0x1bf6, 0x6d79, 0xc6fd, 0x0bb5,
0x2416, 0x2c36, 0x2c36, 0x2c57, 0x0905, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2331, 0x2c77, 0x00c4,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0020, 0x0020, 0x4a69, 0x5aeb, 0x0020, 0x0020, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0041, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0020, 0x10a2, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0905, 0x3457, 0x1bf5, 0x6538, 0xb6bc,
0x7dda, 0xcefd, 0x1bf5, 0x5518, 0x3456, 0x2416, 0x3c77, 0xf7df, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xae7c, 0x1bf5, 0x3456, 0x2c56, 0x1c16, 0x7d99, 0xe77e, 0xbedc,
0x5d18, 0x2416, 0x2c77, 0x2310, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0062, 0x2c15, 0x23b3, 0x0021,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2104, 0x4a49, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0062, 0x0905, 0x0041, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2310, 0x3477, 0x1bf5, 0x8dfa,
0xe77e, 0x6538, 0x2416, 0x2416, 0x2c36, 0x3456, 0x1bf5, 0xae7c, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x5d18, 0x1bf5, 0x2c56, 0x2c36, 0x2c36, 0x2416, 0x2c56, 0x961b,
0x9e1b, 0x2416, 0x2c77, 0x0905, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0841, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x08e5, 0x2c98, 0x2310, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0861, 0x18e3, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0062, 0x0947, 0x0905, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x10a2,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0905, 0x2c77, 0x2c36, 0x1c16,
0x2416, 0x2416, 0x3456, 0x3456, 0x2c36, 0x3456, 0x1bf5, 0x5d18, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xdf5e, 0x2416, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x13d5,
0x23f5, 0x3477, 0x2330, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0841, 0x4a69, 0x0020,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0926, 0x34b8, 0x1acf, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0905, 0x1188, 0x0062, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4a69,
0x0841, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2330, 0x2c77, 0x2c36,
0x2c36, 0x1bf6, 0x2416, 0x2416, 0x2416, 0x2c36, 0x2c36, 0x2416, 0xdf5e, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x8dfa, 0x1bf5, 0x2c36, 0x2c36, 0x2416, 0x3456, 0x3476, 0x2416, 0x2c36,
0x2c36, 0x2c77, 0x0947, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x738e, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0946, 0x34b8, 0x1ace, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x00a3, 0x122b, 0x08e5, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6b6d,
0x2104, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0947, 0x2c77, 0x2416,
0x3c77, 0x6d79, 0x5d18, 0x5d18, 0x54f8, 0x3c77, 0x2c36, 0x1bf5, 0x8dfa, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x54f8, 0x2416, 0x3456, 0x2416, 0x4cd7, 0xcefd, 0xbedc, 0x44b7, 0x3456,
0x2c57, 0x23b3, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x10a2, 0x39c7, 0x5acb, 0xbdd7, 0x632c,
0x4208, 0x2104, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0946, 0x2cd9, 0x1b30, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0021, 0x120a, 0x11ca,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0861, 0x39c7, 0x632c, 0xc638,
0x9cd3, 0x630c, 0x3186, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x2bb3, 0x2457,
0x4497, 0xbebc, 0xf7bf, 0xbedc, 0xb69c, 0xef9e, 0x4cb7, 0x1bf5, 0x54f8, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xdf5e, 0x2416, 0x2c36, 0x2c56, 0x1bf5, 0x9e1b, 0xa65b, 0x7d99, 0x961b, 0x6538,
0x2457, 0x1a2b, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1082, 0x2965, 0x4228, 0xb5b6, 0x528a,
0x3186, 0x18c3, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x08a4, 0xa9ea, 0xc0a3, 0x8800,
0x5000, 0x0800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x11a9,
0x124c, 0x0041, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0841, 0x2945, 0x3186, 0x9492,
0x5acb, 0x2124, 0x18c3, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x122b, 0x3498,
0x2c36, 0x0bb5, 0x8dda, 0x7dba, 0xbebc, 0xae7c, 0x2c56, 0x2c36, 0x2416, 0xdf5e, 0xffff, 0xffff,
0xf7df, 0xffff, 0x9e3b, 0x1bf5, 0x3456, 0x2c36, 0x2416, 0x54f8, 0xef9e, 0xdf5e, 0xbedc, 0x2c36,
0x2c36, 0x08c4, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6b6d, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1061, 0xa882, 0xe800, 0xe000, 0xf800,
0xb20b, 0x1167, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0926, 0x1acf, 0x0083, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x5acb,
0x2104, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x00c4, 0x2c36,
0x2c36, 0x2416, 0x85da, 0xffff, 0x85da, 0x1bf5, 0x2c36, 0x3456, 0x1bf5, 0x9e3b, 0xffff, 0xffdf,
0xffff, 0xffff, 0x5d39, 0x1bf5, 0x2c56, 0x2c36, 0x2c36, 0x2416, 0x44b7, 0x6d79, 0x3c97, 0x2437,
0x2372, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4228, 0x1082,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x5acb, 0xfefb, 0xe535, 0xeb6d, 0x9a4b,
0x2478, 0x0967, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x00a3, 0x2310, 0x0905, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x31a6,
0x1082, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2372,
0x2437, 0x44b7, 0xd73d, 0x4cd8, 0x13d5, 0x3456, 0x2c36, 0x3456, 0x1bf5, 0x5d39, 0xffff, 0xffdf,
0xffff, 0xef9e, 0x3c77, 0x2436, 0x2c36, 0x2c36, 0x2c36, 0x3456, 0x2c36, 0x1bf5, 0x2415, 0x3498,
0x122b, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0861, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x5aeb, 0xffff, 0xf7ff, 0xffff, 0x4d59,
0x1c37, 0x1167, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0082, 0x2330, 0x1188, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x120b,
0x2c98, 0x2c36, 0x2c36, 0x2416, 0x3456, 0x2c36, 0x2c36, 0x2c36, 0x2436, 0x3c77, 0xf7bf, 0xffff,
0xffff, 0xcf1d, 0x1c16, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0x1bf5, 0x13d5, 0x2416, 0x2c36, 0x2c57,
0x0905, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x630c, 0xffff, 0xffff, 0xffff, 0x5d38,
0x1bf5, 0x0905, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0041, 0x2331, 0x120b, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0905,
0x2c57, 0x2c36, 0x2c36, 0x3456, 0x3456, 0x2c36, 0x3c77, 0x4497, 0x2c36, 0x1c16, 0xcf3d, 0xffff,
0xffff, 0xa63b, 0x1bf5, 0x3456, 0x2416, 0x2436, 0x5cf8, 0x7579, 0x963b, 0x8dfa, 0x2436, 0x2bd4,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x18e3, 0x7bcf, 0xbdf7, 0xffff, 0x5539,
0x0126, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0020, 0x4228, 0x1082, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x2331, 0x1a4c, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x23b3, 0x2c57, 0x2c36, 0x2416, 0x1bf5, 0x0395, 0x5d18, 0xae7b, 0x2416, 0x1bf6, 0xa65b, 0xffff,
0xffff, 0x6d59, 0x2416, 0x1bf6, 0x7599, 0xefbf, 0xae7b, 0xdf3d, 0xd73e, 0x5d18, 0x2457, 0x22ef,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3b0e, 0x44f9,
0x0926, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x2124, 0x8430, 0x39e7, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x18e3, 0x2945, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2331, 0x1a6d, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x1aef, 0x2c77, 0x2c36, 0x54f8, 0x75ba, 0x8dfa, 0xcefd, 0xcf1d, 0x2436, 0x1bf5, 0x6d59, 0xffff,
0xffdf, 0x4cd8, 0x2416, 0x2416, 0x5518, 0xdf3e, 0x9e3b, 0xae7b, 0x3cb7, 0x0b94, 0x3498, 0x11c9,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x00a3, 0x2c78,
0x2331, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x18c3, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0841, 0x5aeb, 0x738e, 0x1082,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2351, 0x1a6d, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x11c9, 0x2c78, 0x3456, 0xc6bc, 0xc6dd, 0xa65b, 0x7d99, 0xdf3d, 0x3c97, 0x1bf5, 0x44d8, 0xffff,
0xdf3d, 0x4497, 0x2416, 0x3456, 0x2416, 0x2416, 0x8dfa, 0xefbf, 0xa67b, 0x2c36, 0x2c57, 0x0926,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2310,
0x2c78, 0x00c4, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1082, 0x18e3, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x2393, 0x1a6c, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0926, 0x2c57, 0x2c36, 0x2c36, 0x13d5, 0x13d5, 0x1bf5, 0x7579, 0x3c97, 0x2416, 0x3c77, 0xf7df,
0xa65b, 0x2c56, 0x2c36, 0x2c36, 0x2c56, 0x2c36, 0x13d5, 0x3477, 0xae5b, 0x4cd8, 0x23f5, 0x00a3,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0926,
0x34b9, 0x1a8d, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0021, 0x2bd4, 0x1a4c, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0083, 0x2c15, 0x2c36, 0x2c36, 0x3456, 0x3456, 0x3456, 0x2416, 0x2c36, 0x2c36, 0x2416, 0xdf7e,
0x85da, 0x2436, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x3456, 0x2416, 0x1bd5, 0x2c57, 0x23b3, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x2351, 0x2c57, 0x0062, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0041, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0021, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0041, 0x2c15, 0x120b,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0020, 0x0020, 0x0020, 0x0021, 0x0021, 0x0020, 0x0000, 0x0020, 0x0020, 0x0000, 0x0000,
0x0000, 0x2393, 0x2c57, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x3456, 0x2c36, 0x3456, 0x13d5, 0xc6fd,
0x6d79, 0x2436, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c77, 0x2351, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0946, 0x1aae, 0x2331, 0x23b3, 0x23b3,
0x2372, 0x2393, 0x1a8d, 0x0946, 0x2351, 0x2330, 0x2372, 0x2372, 0x2351, 0x1af0, 0x1aae, 0x11a8,
0x08c4, 0x34b9, 0x1a8d, 0x0000, 0x0000, 0x0000, 0x0000, 0x124c, 0x23b3, 0x2372, 0x08e5, 0x0000,
0x0000, 0x0000, 0x0000, 0x120b, 0x2bd4, 0x2393, 0x08c4, 0x0000, 0x0000, 0x0000, 0x0000, 0x11a9,
0x23b3, 0x0083, 0x0000, 0x0000, 0x0000, 0x0000, 0x1a8d, 0x2393, 0x2372, 0x00a3, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x11ea, 0x2351, 0x23b4, 0x2bd4, 0x2bd4, 0x2393, 0x2372, 0x2c77,
0x1188, 0x0000, 0x0000, 0x0000, 0x0000, 0x1aef, 0x23b3, 0x2331, 0x0021, 0x0000, 0x0000, 0x0000,
0x0947, 0x2c16, 0x2bf5, 0x2bf5, 0x23b3, 0x2393, 0x23b4, 0x2bf5, 0x2bf5, 0x2bd4, 0x00a3, 0x0000,
0x0000, 0x2351, 0x2c77, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x1bf5, 0xa65b,
0x6538, 0x2436, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c77, 0x1acf, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x11c9, 0x2c77, 0x3498, 0x3498, 0x2372, 0x1a6d,
0x1a8d, 0x2bd4, 0x2372, 0x11ea, 0x34b9, 0x2c77, 0x2351, 0x2330, 0x23b3, 0x2c98, 0x3498, 0x3498,
0x122b, 0x1aae, 0x3498, 0x0083, 0x0000, 0x0000, 0x0041, 0x2bf5, 0x2c77, 0x34b8, 0x1a6c, 0x0000,
0x0000, 0x0000, 0x0000, 0x1a8d, 0x2c98, 0x3498, 0x2330, 0x0000, 0x0000, 0x0000, 0x0000, 0x122b,
0x3498, 0x08a4, 0x0000, 0x0000, 0x0000, 0x0062, 0x2c36, 0x2c77, 0x34b8, 0x11ea, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x1a8d, 0x34b9, 0x2c16, 0x2310, 0x1a4c, 0x1a6d, 0x2393, 0x23b3, 0x11ca,
0x2c78, 0x0905, 0x0000, 0x0000, 0x08e4, 0x2c78, 0x2c57, 0x3498, 0x0947, 0x0000, 0x0000, 0x0000,
0x00a3, 0x1a4c, 0x1a4c, 0x122b, 0x2bd4, 0x3498, 0x2372, 0x120b, 0x1a6c, 0x122c, 0x0041, 0x0000,
0x0000, 0x1acf, 0x2c77, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x1bf6, 0x95fa,
0x54d8, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c78, 0x1a4c, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x08e5, 0x2c57, 0x2c57, 0x2bf5, 0x0926, 0x0000, 0x0000,
0x0000, 0x0000, 0x0062, 0x11c9, 0x2c57, 0x2c36, 0x00a3, 0x0000, 0x0000, 0x08c4, 0x23b3, 0x2c57,
0x2c36, 0x1188, 0x2c56, 0x1aef, 0x0000, 0x0000, 0x0947, 0x2c77, 0x2bf5, 0x2c57, 0x23b3, 0x0000,
0x0000, 0x0000, 0x0000, 0x1a6d, 0x2c77, 0x2c15, 0x2c98, 0x0968, 0x0000, 0x0000, 0x0000, 0x120b,
0x2c77, 0x00a3, 0x0000, 0x0000, 0x0000, 0x11a8, 0x2c77, 0x2bf4, 0x2c57, 0x2372, 0x0000, 0x0000,
0x0000, 0x0000, 0x00a3, 0x2c36, 0x2c36, 0x00c4, 0x0000, 0x0000, 0x0000, 0x0000, 0x08c4, 0x0000,
0x122b, 0x2c36, 0x0082, 0x0000, 0x1a4c, 0x2c57, 0x2bf5, 0x2c77, 0x1acf, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x2331, 0x34b9, 0x1a8d, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x1a4c, 0x2c78, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0x7559,
0x4497, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c16, 0x3498, 0x11ea, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1aef, 0x2c77, 0x2c57, 0x0967, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x11a9, 0x2c77, 0x2c36, 0x08c4, 0x0000, 0x0000, 0x0000, 0x0947, 0x2c57,
0x3498, 0x0967, 0x11a9, 0x34d9, 0x0905, 0x0000, 0x1b10, 0x23b3, 0x122b, 0x2c77, 0x2c57, 0x0926,
0x0000, 0x0000, 0x0000, 0x1a6d, 0x2c78, 0x2c16, 0x2c57, 0x2bd4, 0x0041, 0x0000, 0x0000, 0x122b,
0x2c77, 0x08a4, 0x0000, 0x0000, 0x0000, 0x2351, 0x2392, 0x1a4c, 0x2c77, 0x2c36, 0x08c4, 0x0000,
0x0000, 0x0000, 0x0947, 0x2c77, 0x2c16, 0x0083, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x2372, 0x23b4, 0x0041, 0x2bd4, 0x1aef, 0x1aae, 0x3498, 0x2bd4, 0x0062, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x2351, 0x34b8, 0x1aae, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x11ea, 0x3498, 0x2c16, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0x5cf8,
0x3456, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c78, 0x1188, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0082, 0x2bf4, 0x2c77, 0x2372, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x11c9, 0x2c77, 0x2c36, 0x08c4, 0x0000, 0x0020, 0x0000, 0x0083, 0x2c36,
0x3498, 0x11c9, 0x0000, 0x23b3, 0x23d4, 0x0041, 0x2c15, 0x1a8e, 0x0041, 0x2c36, 0x2c98, 0x1a6d,
0x0000, 0x0000, 0x0000, 0x1a6d, 0x2c77, 0x2c36, 0x2c36, 0x2c98, 0x1a8d, 0x0000, 0x0000, 0x120b,
0x2c77, 0x08a4, 0x0000, 0x0000, 0x0041, 0x2c77, 0x122b, 0x0083, 0x2c57, 0x2c78, 0x122b, 0x0000,
0x0000, 0x0000, 0x0905, 0x2c57, 0x2c57, 0x2372, 0x0041, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0062, 0x2c15, 0x2351, 0x2c77, 0x0967, 0x0946, 0x2c78, 0x2c77, 0x0967, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x2351, 0x34b9, 0x1aae, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0988, 0x2c78, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x4497,
0x3456, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c16, 0x2c98, 0x0947, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0905, 0x2c56, 0x2c78, 0x122b, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x11c9, 0x2c77, 0x2c36, 0x08c4, 0x0000, 0x0000, 0x0000, 0x120a, 0x2c57,
0x2c77, 0x0967, 0x0000, 0x08e4, 0x2c57, 0x2352, 0x2c77, 0x0926, 0x0000, 0x2331, 0x2c98, 0x23d4,
0x0020, 0x0000, 0x0000, 0x1a8d, 0x2bf4, 0x1acf, 0x2c77, 0x2c36, 0x2c57, 0x08e4, 0x0000, 0x122b,
0x2c77, 0x08a3, 0x0000, 0x0000, 0x11a8, 0x34b8, 0x08e4, 0x0000, 0x2372, 0x2c98, 0x2393, 0x0000,
0x0000, 0x0000, 0x0000, 0x1b10, 0x3498, 0x2c57, 0x23b3, 0x08e5, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0946, 0x2c78, 0x2bf5, 0x0041, 0x0041, 0x2bf5, 0x2c78, 0x1aef, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x2351, 0x34b9, 0x1aae, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0947, 0x2c98, 0x2c16, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x3457,
0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c78, 0x0947, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0926, 0x2c57, 0x2c78, 0x0967, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x11a9, 0x2c77, 0x2c36, 0x0926, 0x00a3, 0x0967, 0x2310, 0x2c78, 0x34b8,
0x2372, 0x0020, 0x0000, 0x0000, 0x11a9, 0x3498, 0x2393, 0x0062, 0x0000, 0x11ea, 0x2c77, 0x2c57,
0x0926, 0x0000, 0x0000, 0x1a8d, 0x2c15, 0x00a3, 0x2bf4, 0x2c56, 0x2c77, 0x2372, 0x0000, 0x122b,
0x2c77, 0x08a4, 0x0000, 0x0000, 0x2310, 0x2bd4, 0x0020, 0x0000, 0x122c, 0x2c78, 0x3436, 0x0105,
0x1000, 0x4000, 0x0800, 0x0021, 0x1af0, 0x2c78, 0x2c57, 0x2c36, 0x11ea, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0020, 0x23b3, 0x2c36, 0x00a3, 0x0000, 0x1aef, 0x2c78, 0x2bf5, 0x0062, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x2351, 0x34b9, 0x1aae, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0947, 0x2c78, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c98, 0x0947, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0926, 0x2c57, 0x2c77, 0x0967, 0x0000, 0x0000, 0x0000, 0x0000,
0x0062, 0x11c9, 0x0926, 0x11c9, 0x2c78, 0x2bf5, 0x2393, 0x2c36, 0x2c98, 0x2c36, 0x2331, 0x120b,
0x0041, 0x0000, 0x0000, 0x0000, 0x0021, 0x2bf5, 0x23d4, 0x0020, 0x0000, 0x0082, 0x2c15, 0x2c98,
0x1a8d, 0x0000, 0x0000, 0x1a8d, 0x2c36, 0x0000, 0x11a8, 0x2c98, 0x2c15, 0x3498, 0x1188, 0x11ea,
0x2c78, 0x00c4, 0x0000, 0x0041, 0x2c57, 0x1a6d, 0x0000, 0x0000, 0x08c4, 0x3436, 0x1cb9, 0x3a4c,
0x8000, 0x1800, 0x0000, 0x0000, 0x0000, 0x1a4c, 0x2c57, 0x2c57, 0x2c98, 0x1af0, 0x0021, 0x0000,
0x0000, 0x0000, 0x08e4, 0x2c36, 0x2c77, 0x2372, 0x0000, 0x0967, 0x2c77, 0x2c77, 0x11a9, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x2351, 0x34b9, 0x1aae, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0947, 0x2c98, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x3456, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c98, 0x0947, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0905, 0x2c36, 0x2c98, 0x1188, 0x0000, 0x0000, 0x0000, 0x0000,
0x0926, 0x34d9, 0x2393, 0x11ea, 0x2c78, 0x2bf4, 0x11c9, 0x2c36, 0x2c36, 0x23b3, 0x0041, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0947, 0x2c57, 0x2c57, 0x1aae, 0x0000, 0x0000, 0x2310, 0x3498,
0x2bd4, 0x0041, 0x0000, 0x1a8d, 0x2c36, 0x0041, 0x0000, 0x2372, 0x2c77, 0x2c56, 0x23b3, 0x1a8d,
0x2c57, 0x08c4, 0x0000, 0x11a9, 0x3498, 0x0905, 0x0000, 0x0000, 0x0020, 0x13b3, 0x6372, 0xa968,
0x1821, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0946, 0x2bf4, 0x2c57, 0x2c98, 0x1aef, 0x0000,
0x0000, 0x0000, 0x1a6d, 0x2c16, 0x11c9, 0x34b8, 0x11ca, 0x0020, 0x2bf5, 0x2c98, 0x2310, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x2351, 0x34b9, 0x1aae, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0947, 0x2c98, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x3457,
0x3456, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x3478, 0x0967, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0083, 0x2bf5, 0x3498, 0x1acf, 0x0000, 0x0000, 0x0000, 0x0000,
0x0926, 0x3498, 0x2351, 0x11e9, 0x2c77, 0x2c36, 0x0062, 0x11a9, 0x2c77, 0x2c77, 0x2330, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x1aef, 0x2bf5, 0x0946, 0x2c78, 0x11ea, 0x0000, 0x11ea, 0x2c77,
0x2c57, 0x0947, 0x0000, 0x1a8d, 0x2c36, 0x0062, 0x0000, 0x0905, 0x2c77, 0x2c36, 0x2c36, 0x2c36,
0x2c16, 0x08c4, 0x0000, 0x2330, 0x2bf5, 0x0000, 0x0000, 0x0000, 0x0000, 0x80e4, 0xd126, 0x3bf5,
0x0126, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0082, 0x2351, 0x2c57, 0x2c77, 0x08e5,
0x0000, 0x0021, 0x23d4, 0x2352, 0x0000, 0x1aef, 0x3498, 0x0062, 0x1a8d, 0x3498, 0x2c15, 0x00a3,
0x0000, 0x0000, 0x0000, 0x0000, 0x2351, 0x34b9, 0x1aae, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x1188, 0x3478, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x4497,
0x4497, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c16, 0x3498, 0x11ea, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2331, 0x2c78, 0x2bf5, 0x0062, 0x0000, 0x0020, 0x0000,
0x0946, 0x34b9, 0x2351, 0x11c9, 0x2c77, 0x2c36, 0x08c4, 0x0000, 0x1a8d, 0x2c77, 0x2c77, 0x1a4c,
0x0000, 0x0000, 0x0000, 0x0021, 0x2c36, 0x1acf, 0x0000, 0x1188, 0x2c78, 0x08c4, 0x0041, 0x2c36,
0x2c98, 0x1ace, 0x0000, 0x1a8d, 0x2c36, 0x0041, 0x0000, 0x0000, 0x1a8d, 0x2c98, 0x2c15, 0x2c56,
0x2c36, 0x00a3, 0x0000, 0x2c77, 0x1a6d, 0x0000, 0x0000, 0x1800, 0xb000, 0xe820, 0x4bb3, 0x24b9,
0x1a6c, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0062, 0x2c36, 0x2c77, 0x11c9,
0x0000, 0x08e4, 0x34b9, 0x11c9, 0x0000, 0x0083, 0x2c78, 0x1aef, 0x0926, 0x2c77, 0x2c77, 0x11a9,
0x0000, 0x0000, 0x0000, 0x0000, 0x2351, 0x34b9, 0x1aae, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x120a, 0x3498, 0x2c16, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0x5cf8,
0x54d8, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c78, 0x1a4c, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0967, 0x2c77, 0x2c57, 0x1af0, 0x0020, 0x0000, 0x0000,
0x0905, 0x34b9, 0x2351, 0x11c9, 0x2c77, 0x2c36, 0x08c4, 0x0000, 0x0020, 0x2351, 0x2c77, 0x2c77,
0x1188, 0x0000, 0x0000, 0x0967, 0x34b9, 0x0947, 0x0000, 0x0000, 0x120b, 0x2c36, 0x0041, 0x1aef,
0x3498, 0x2bf4, 0x0000, 0x1a6d, 0x2c36, 0x0041, 0x0000, 0x0000, 0x0062, 0x2c15, 0x2c56, 0x2c36,
0x2c36, 0x0062, 0x0968, 0x34b9, 0x0905, 0x0000, 0x3000, 0xd800, 0xf800, 0x4821, 0x1351, 0x3498,
0x23b3, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0905, 0x2c36, 0x2c57, 0x08e4,
0x0000, 0x1a6d, 0x2c77, 0x0083, 0x0000, 0x0000, 0x1a6c, 0x34b9, 0x0968, 0x2372, 0x3498, 0x2330,
0x0000, 0x0000, 0x0000, 0x0000, 0x2351, 0x34b8, 0x1aae, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x1a6d, 0x2c78, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0x7559,
0x6538, 0x2436, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c77, 0x1acf, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1a8d, 0x3498, 0x2c98, 0x2372, 0x11c9, 0x0967,
0x120a, 0x3498, 0x2351, 0x11c9, 0x2c77, 0x2c36, 0x08c4, 0x0000, 0x0000, 0x0062, 0x23b3, 0x2c77,
0x2c15, 0x00a3, 0x0000, 0x1acf, 0x2c56, 0x0041, 0x0000, 0x0000, 0x0000, 0x1acf, 0x2372, 0x1188,
0x2c77, 0x2c78, 0x0926, 0x1a4c, 0x2c36, 0x0041, 0x0000, 0x0000, 0x0000, 0x11ea, 0x2c98, 0x2c36,
0x2c36, 0x0083, 0x1aef, 0x2c15, 0x0000, 0x0000, 0x9000, 0xf800, 0x6000, 0x0000, 0x120a, 0x2c78,
0x2c57, 0x08e5, 0x08e5, 0x2372, 0x11a9, 0x0926, 0x08c4, 0x0967, 0x23d4, 0x34b9, 0x1aef, 0x0000,
0x0021, 0x23b4, 0x2352, 0x0000, 0x0000, 0x0000, 0x0062, 0x2c57, 0x2331, 0x1a8e, 0x3498, 0x2c15,
0x0083, 0x0000, 0x0000, 0x0000, 0x2351, 0x34b9, 0x1aae, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x1acf, 0x2c77, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x1bf6, 0x95fa,
0x6d79, 0x2436, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c77, 0x2351, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x120a, 0x2bd4, 0x2c98, 0x34d9, 0x3498,
0x2c77, 0x34b8, 0x2372, 0x11c9, 0x34b8, 0x2c77, 0x08c4, 0x0000, 0x0000, 0x0000, 0x00a3, 0x2c15,
0x34b9, 0x23b4, 0x00a3, 0x2c36, 0x1aef, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1b10, 0x2372,
0x2393, 0x3498, 0x1aae, 0x1a6c, 0x2c57, 0x0041, 0x0000, 0x0000, 0x0000, 0x0000, 0x2352, 0x3498,
0x2c15, 0x0946, 0x2c57, 0x1a8e, 0x0000, 0x0000, 0x1000, 0x1800, 0x0000, 0x0000, 0x0083, 0x2c36,
0x3498, 0x124c, 0x0905, 0x34d9, 0x34b8, 0x2c78, 0x2c57, 0x3498, 0x2c98, 0x2331, 0x0041, 0x0000,
0x0905, 0x34b9, 0x11a9, 0x0000, 0x0000, 0x0000, 0x0000, 0x1a4c, 0x3498, 0x1aae, 0x2c15, 0x3498,
0x11ca, 0x0000, 0x0000, 0x0000, 0x2351, 0x3498, 0x1aae, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x2351, 0x2c77, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x3436, 0x1bf5, 0xa65b,
0x85da, 0x2436, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c57, 0x2393, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0041, 0x00a3, 0x00a3, 0x00a3,
0x08c4, 0x08c4, 0x0083, 0x0041, 0x08c4, 0x08c4, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0062,
0x08c4, 0x08e4, 0x0083, 0x00a3, 0x0041, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2351,
0x2331, 0x0083, 0x0082, 0x0062, 0x00a3, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0041, 0x08c4,
0x00a3, 0x0041, 0x08c4, 0x0021, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0083,
0x00a3, 0x0062, 0x0021, 0x00a3, 0x00a3, 0x00a4, 0x08c4, 0x00c4, 0x00a4, 0x0000, 0x0000, 0x0000,
0x0021, 0x08c4, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0062, 0x2c36, 0x2393, 0x00a3, 0x00a3,
0x0062, 0x0000, 0x0000, 0x0000, 0x0062, 0x00a3, 0x0062, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x23b3, 0x2c57, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x3456, 0x13d5, 0xc6fd,
0xa65b, 0x2c56, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c56, 0x2c15, 0x0083,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020,
0x2331, 0x1a4c, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1a8d, 0x34b9, 0x0926, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x00a3, 0x2c15, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0xe77e,
0xdf3d, 0x4497, 0x2416, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c57, 0x0926,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0020, 0x2351, 0x1a4c, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0905, 0x3498, 0x2331, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0946, 0x2c77, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0x3c77, 0xf7df,
0xffdf, 0x4cd8, 0x2416, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c16, 0x3498, 0x11a9,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0020, 0x2331, 0x122b, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2372, 0x2c77, 0x00c4,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x11ca, 0x3498, 0x2c16, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0x4cf8, 0xffff,
0xffff, 0x6d59, 0x1bf5, 0x2c56, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c77, 0x1aef,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x1acf, 0x1a4c, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x122b, 0x34d9, 0x11ea,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x1aef, 0x2c77, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c56, 0x1bf5, 0x6d59, 0xffff,
0xffff, 0xa63b, 0x1bf5, 0x3456, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c57, 0x23b3,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x1a8d, 0x1a6c, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x08e5, 0x3498, 0x2351,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x18c3, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x23b4, 0x2c57, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x3456, 0x1bf5, 0xa65b, 0xffff,
0xffff, 0xc6fd, 0x1bf6, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c57,
0x0905, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x10a2, 0x18c3, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x11ea, 0x1a8d, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0021, 0x23d4, 0x2c36,
0x0083, 0x0000, 0x0000, 0x0000, 0x2104, 0x8410, 0x2945, 0x0000, 0x0000, 0x0000, 0x0000, 0x0905,
0x2c57, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x3436, 0x1bf6, 0xcf3d, 0xffff,
0xffff, 0xefbf, 0x3c77, 0x2436, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c16, 0x3498,
0x120b, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0020, 0x39c7, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x2965, 0x4a69, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1188, 0x1a8d, 0x0062, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1aef, 0x34b9,
0x0947, 0x0000, 0x0000, 0x0000, 0x0861, 0x4228, 0x1082, 0x0000, 0x0000, 0x0000, 0x0000, 0x122b,
0x3498, 0x2c16, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2436, 0x3c97, 0xf7bf, 0xffff,
0xffff, 0xffff, 0x5d38, 0x1bf5, 0x2c56, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c57,
0x2372, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x31a6, 0x8430, 0x2124, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x4208, 0x6b4d, 0x0000, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0905, 0x1a8d, 0x00a3, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1a4c, 0x34d9,
0x11c9, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2372,
0x2c57, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c56, 0x1bf5, 0x6559, 0xffff, 0xffdf,
0xf7df, 0xffff, 0x9e3b, 0x1bf5, 0x2c56, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x2c36, 0x00c4, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x2124, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0841,
0x4208, 0x630c, 0xad96, 0xc618, 0x6b6d, 0x4a69, 0x1082, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0041, 0x1a6c, 0x0926, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0841,
0x630c, 0x39c7, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x11ea, 0x34b9,
0x1aae, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x08c4, 0x2c56,
0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x3456, 0x1bf5, 0x9e3b, 0xffff, 0xffdf,
0xffff, 0xffff, 0xdf5e, 0x2416, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x2c98, 0x122b, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x7bef, 0xad75, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x11a8, 0x11a8,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x18e3, 0x6b4d,
0xad55, 0x31a6, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0926, 0x34b8,
0x1b10, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x122b, 0x2c98,
0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0xdf5e, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x54f8, 0x2416, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x2c57, 0x23b3, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x2124, 0x4a49, 0x7bcf, 0x3186, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0905,
0x1188, 0x0062, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x18e3, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x18c3, 0x4a69, 0x528a, 0x630c,
0x10a2, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x08c4, 0x34b8,
0x2310, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x23b3, 0x2c57,
0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0x54f8, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x8dfa, 0x1bf5, 0x2c56, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x3456, 0x2c36,
0x2c36, 0x2c77, 0x0947, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x52aa, 0x1082, 0x2945, 0x52aa, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0041, 0x0967, 0x00c4, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2104, 0x8c51, 0x31a6, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0861, 0x4a49, 0x52aa, 0x18c3, 0x4a49, 0x2945,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0947, 0x34b9,
0x1aef, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0967, 0x2c77, 0x2c36,
0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c56, 0x1bf5, 0x961b, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xdf5e, 0x2416, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x1bf5, 0x1bf6,
0x2c36, 0x2c77, 0x2330, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x3186, 0x39e7, 0x0000, 0x0000, 0x39e7, 0x2965, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x00c4, 0x08e5, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x4208, 0x0841, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0841, 0x4208, 0x5acb, 0x2945, 0x0000, 0x3186, 0x4208, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x11ea, 0x34b9,
0x1a4c, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2330, 0x2c77, 0x2c36,
0x2c36, 0x2c36, 0x3456, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0xdf5e, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x5d18, 0x1bf5, 0x2c56, 0x2c36, 0x2416, 0x2416, 0x7db9, 0xae7b,
0x1c16, 0x2c36, 0x2c57, 0x0905, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020,
0x5acb, 0x0020, 0x0000, 0x0000, 0x0841, 0x528a, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0041, 0x0082, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x39c7, 0x5aeb, 0x31a6, 0x0000, 0x0000, 0x2124, 0x5acb, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1a8d, 0x34d9,
0x0967, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0905, 0x2c77, 0x2c36, 0x2416,
0x1c16, 0x1c16, 0x1bd5, 0x2c36, 0x2c36, 0x2c56, 0x1bf5, 0x5d18, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xae7c, 0x1bf5, 0x3456, 0x2c36, 0x3477, 0xd71d, 0xdf3d, 0xb69c,
0x5d18, 0x23f5, 0x3477, 0x2310, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3186,
0x39c7, 0x0000, 0x0000, 0x0000, 0x0000, 0x4228, 0x2124, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x2965, 0x5acb, 0x39e7, 0x0841, 0x0000, 0x0000, 0x0841, 0x5acb, 0x0841, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2393, 0x2c36,
0x0062, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2310, 0x3477, 0x2415, 0x3477,
0xcf1d, 0xd73e, 0x95fa, 0x3456, 0x1bf5, 0x3456, 0x1bf5, 0xae7c, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xf7df, 0x3c97, 0x2c36, 0x1bf6, 0x5d39, 0xc6dc, 0x7dba, 0xae7c,
0xae7c, 0x1bf5, 0x2c36, 0x2c57, 0x0905, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x5acb,
0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x10a2, 0x52aa, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2124,
0x52aa, 0x4228, 0x0861, 0x0000, 0x0000, 0x0020, 0x0000, 0x5acb, 0x18c3, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0926, 0x34b9, 0x1a6d,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0905, 0x2c57, 0x2c36, 0x1bf5, 0xb69c,
0x85ba, 0x54f8, 0xc6dd, 0xef9e, 0x8dfa, 0x1bf5, 0x3c97, 0xf7df, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x8dfa, 0x13f5, 0x3456, 0x2c36, 0x9e1b, 0x5518, 0x85da,
0x3c97, 0x2c36, 0x2c36, 0x2c77, 0x2351, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x31a6, 0x39c7,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x528a, 0x18c3, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x18e3, 0x52aa, 0x4a69,
0x10a2, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4228, 0x39c7, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2372, 0x2c36, 0x0062,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2331, 0x2c77, 0x2c36, 0x2416, 0xd71d,
0x6d59, 0x0394, 0x0bb5, 0x4cd8, 0x6d59, 0x0bd5, 0x8e1b, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xe77e, 0x2c56, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x1bf5,
0x2c36, 0x2c36, 0x2416, 0x2c36, 0x2c78, 0x11a9, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x10a2, 0x10a2, 0x52aa, 0x0020,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x18e3, 0x4a49, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x10a2, 0x4a69, 0x52aa, 0x18e3, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3186, 0x4228, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1a6d, 0x2c77, 0x0905, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x11a8, 0x2c77, 0x2c36, 0x2c36, 0x2416, 0x54f8,
0xe77e, 0xc6dd, 0x5d38, 0x2416, 0x1bf5, 0x2c56, 0xe79e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x7dba, 0x13f5, 0x3456, 0x2c36, 0x2c36, 0x3456,
0x1bf5, 0x2c56, 0x4497, 0x2416, 0x2c57, 0x2bf5, 0x0062, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x39c7, 0x6b4d, 0x2965, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x528a, 0x0861, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0841, 0x39e7, 0x52aa, 0x2945, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x10a2, 0x630c, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1a8d, 0x2bf4, 0x08e4, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0062, 0x2bf4, 0x2c57, 0x2c36, 0x2c36, 0x3456, 0x1c16,
0x2c56, 0x85da, 0xe77e, 0x7d99, 0x0bb5, 0x85da, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xe77e, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0x13f5,
0x6538, 0xe77e, 0xc6dd, 0x1bf5, 0x2c36, 0x2c77, 0x1aae, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x52aa, 0x8410, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2945, 0x39e7, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0020, 0x39c7, 0x5acb, 0x3186, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x52aa, 0x0861, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0021, 0x0905, 0x1a8d, 0x122c, 0x0041, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1aae, 0x2c77, 0x2c36, 0x2c36, 0x1bf6, 0x2416, 0x3456,
0x2c36, 0x13d5, 0x2c36, 0x3456, 0x2436, 0xe77e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x85da, 0x1bf5, 0x2c56, 0x44b7, 0xc6fd,
0xef9e, 0x4cd8, 0x8dfa, 0x9e3b, 0x13d5, 0x3436, 0x2c77, 0x0967, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0861, 0x4228, 0x6b4d, 0xbdf7, 0xbdd7, 0x73ae, 0x31a6,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x52aa, 0x0020, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x2124, 0x52aa, 0x31a6, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0861, 0x0000, 0x528a, 0x2945, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x00a3, 0x0926, 0x0041, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0967, 0x2c77, 0x2c36, 0x2416, 0x5d18, 0xc6fd, 0x44b7, 0x13d5,
0x2c36, 0x3456, 0x3456, 0x13d5, 0x85da, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xefbf, 0x3477, 0x2c56, 0xd71d, 0x9e3b,
0xc6dc, 0x3456, 0xb6dc, 0xae7c, 0x13d5, 0x3436, 0x2c57, 0x2bf5, 0x0083, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1082, 0x10a2, 0x632c, 0x52aa, 0x4a69, 0x5aeb,
0x2945, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x31a6, 0x39c7, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x31a6, 0x9492, 0x6b4d, 0x18c3, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x10a2, 0x31a6, 0x3186, 0x4228, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0083, 0x2bf5, 0x2c56, 0x2c36, 0x2c36, 0x1c16, 0x8dda, 0xef9e, 0x8dfa,
0x2436, 0x2c36, 0x2c36, 0x3477, 0xef9f, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xae7c, 0x1bf5, 0x3456, 0x13b5,
0x7dba, 0xd73d, 0xa65b, 0x2c36, 0x2c36, 0x3456, 0x2c36, 0x2c77, 0x2351, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4208, 0x2124, 0x0000, 0x1082,
0x4a49, 0x528a, 0x39c7, 0x0841, 0x0000, 0x0000, 0x0000, 0x0000, 0x0841, 0x5acb, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x4a49, 0x630c, 0x3186, 0x4228, 0x4a69, 0x4a69, 0x4208, 0x4208, 0x2124, 0x0861, 0x0020, 0x0000,
0x0000, 0x7bef, 0x3186, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x2351, 0x2c77, 0x2c36, 0x1bf5, 0x8dfa, 0xc6dd, 0x0bd5, 0x7dda, 0xb69c,
0x2416, 0x3456, 0x13f5, 0xae7c, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x54f8, 0x1bf6, 0x3456,
0x2416, 0x2c36, 0x1bf5, 0x2c36, 0x2c36, 0x1bf5, 0x2436, 0x2c36, 0x2c98, 0x1a6d, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2104, 0x10a2, 0x0000, 0x0000,
0x0000, 0x0861, 0x39c7, 0x5aeb, 0x4208, 0x0861, 0x0000, 0x0000, 0x0000, 0x39c7, 0x2965, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0861, 0x18e3, 0x18e3, 0x39e7, 0x4a69, 0x738e, 0x7bcf,
0x73ae, 0xce79, 0x52ab, 0x39e7, 0x1082, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x1a6d, 0x2c98, 0x2c16, 0x3456, 0x2c36, 0x3456, 0x961b, 0xef9e, 0xd73d, 0x2c36,
0x2c36, 0x2416, 0x54f8, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xdf3e, 0x2436, 0x2c36,
0x2c36, 0x2c36, 0x3456, 0x2c16, 0x2416, 0xae7c, 0x5d18, 0x1bd5, 0x3436, 0x2c98, 0x11ea, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x3186, 0x52aa, 0x4228, 0x18e3, 0x0000, 0x0020, 0x5acb, 0x39e7,
0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1082, 0x2986,
0x632d, 0x94b3, 0x2966, 0x2124, 0x0841, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x11ea, 0x2c98, 0x2c36, 0x2c36, 0x1bf5, 0x2c36, 0x2c36, 0x13d5, 0x44b7, 0x44b7, 0x2436,
0x2c36, 0x2436, 0xdf3e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x961b, 0x13f5,
0x3456, 0x2c36, 0x2c36, 0x3456, 0xcf1d, 0xc6fd, 0xa65b, 0x4cd8, 0x2416, 0x2c36, 0x2c77, 0x1188,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2124, 0x528a, 0x528a, 0x18c3, 0x39c7, 0x8430,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0841, 0x5aeb, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x1188, 0x2c77, 0x2c36, 0x2416, 0x3c77, 0x8dfa, 0x2436, 0x2416, 0x3456, 0x2416, 0x2416, 0x3456,
0x13d5, 0x961b, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x54f8,
0x1bf6, 0x3456, 0x1bd5, 0x85da, 0xcefd, 0x7d99, 0xe77e, 0x85ba, 0x1bf5, 0x3436, 0x2c36, 0x2c56,
0x0946, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3186, 0x738e, 0x9492, 0xc638,
0x4228, 0x2965, 0x10a2, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x1082, 0x39e7, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0946,
0x2c56, 0x2c36, 0x2c36, 0x2c36, 0x2c56, 0xbebc, 0xd73d, 0x3c97, 0x2416, 0x2c36, 0x3456, 0x1c16,
0x4cf8, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xdf7e,
0x2c56, 0x2c36, 0x2416, 0x4497, 0xb69c, 0x6d59, 0x5d59, 0x2c36, 0x2c36, 0x2c36, 0x2c16, 0x2c56,
0x2c36, 0x0946, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x10a2, 0x3186, 0x5aeb, 0xbdd7,
0x5aeb, 0x39e7, 0x18e3, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0841, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0946, 0x2c36,
0x2c56, 0x2c36, 0x2c56, 0x3456, 0x2c36, 0x13d5, 0x95fa, 0xcf1d, 0x3456, 0x3456, 0x2c36, 0x2c36,
0xdf5e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xb6bc, 0x13f5, 0x3456, 0x2416, 0x4497, 0x3c77, 0x13d5, 0x2c36, 0x2416, 0x3456, 0x3477, 0x2c36,
0x2c56, 0x2c56, 0x1188, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x738e,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1188, 0x2c56, 0x2c56,
0x2c36, 0x2416, 0x1bf6, 0x1bf5, 0x2c36, 0x2c56, 0x1bf5, 0x3456, 0xae7b, 0x5cf8, 0x0bd5, 0xb6bc,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0x8dfa, 0x13d5, 0x3456, 0x2416, 0x2c36, 0x3456, 0x2416, 0x3c77, 0xe77e, 0x5d18, 0x0bb5,
0x2c36, 0x2c56, 0x2c57, 0x11ea, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4a69,
0x1082, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x11ea, 0x2c57, 0x2c56, 0x2c36,
0x23f6, 0x7dba, 0xbebc, 0x6559, 0x1c16, 0x2c36, 0x3456, 0x2416, 0x3c97, 0x1bf5, 0x8dfa, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x6d59, 0x1bf5, 0x3456, 0x2c36, 0x2c36, 0x3477, 0xdf5e, 0x85ba, 0x1bf6, 0x85da,
0x2c56, 0x2c36, 0x2c36, 0x2c98, 0x1a6d, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0861,
0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1a6d, 0x2c98, 0x2c36, 0x2c36, 0x3456,
0x1bd5, 0x961b, 0xd73d, 0xb6bc, 0xcefd, 0x4497, 0x2416, 0x3456, 0x13d5, 0x6d59, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xf7df, 0x4cd8, 0x1bf5, 0x3456, 0x1bf6, 0x6538, 0xe77e, 0x2416, 0xcefd, 0xbedc,
0x2416, 0x2c36, 0x2c36, 0x2c36, 0x2c98, 0x2351, 0x0083, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0083, 0x2351, 0x2c98, 0x2c36, 0x2c36, 0x2c36, 0x2c56,
0x2c36, 0x3456, 0xcf1d, 0x3497, 0x54f8, 0x3477, 0x2c36, 0x1bf5, 0x4cd8, 0xf7df, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xef9e, 0x44b7, 0x1c16, 0x3456, 0x2416, 0x6d59, 0xf7bf, 0xb69c, 0x13f5,
0x2c36, 0x2c36, 0x1bf5, 0x2416, 0x2c36, 0x2c77, 0x2bf5, 0x0967, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x1188, 0x2bf5, 0x2c77, 0x2c36, 0x2416, 0x1c16, 0x2416, 0x2416,
0x3456, 0x13d5, 0x961b, 0x8dfa, 0x0bb5, 0x3456, 0x1c16, 0x44b7, 0xef9e, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xe77e, 0x3c97, 0x1c16, 0x3456, 0x1bf5, 0x5d18, 0x2436, 0x2c36,
0x2c36, 0x1c16, 0xa65b, 0x54f8, 0x13f5, 0x2c36, 0x2c56, 0x2c77, 0x1a8d, 0x0062, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x18e3, 0x39c7, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0062, 0x1aae, 0x2c77, 0x2c56, 0x2c36, 0x2416, 0x4cd8, 0xbebc, 0xc6fd, 0x54d8,
0x2416, 0x2c36, 0x2c36, 0x2c36, 0x3456, 0x1c16, 0x3c97, 0xe77e, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xe77e, 0x3c97, 0x1c16, 0x3456, 0x2416, 0x2c36, 0x3456,
0x1bf5, 0x961b, 0xdf5e, 0xa65b, 0x85ba, 0x2416, 0x2c36, 0x2c36, 0x2c77, 0x2bd4, 0x11a9, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020,
0x52aa, 0x73ae, 0x10a2, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x11c9, 0x2bf5, 0x2c78, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0x9e1b, 0x6538, 0xd71d, 0xcf1d,
0x1c16, 0x2c36, 0x2c36, 0x3456, 0x1c16, 0x3c97, 0xe77e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xe77e, 0x3cb7, 0x1bf5, 0x3456, 0x2c36, 0x2416,
0x4cd7, 0xef7e, 0x85ba, 0xcefd, 0xa65b, 0x1bf5, 0x2c36, 0x2c36, 0x2c36, 0x2c57, 0x2c77, 0x2331,
0x0926, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0861, 0x1082, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0926,
0x2351, 0x2c78, 0x2c57, 0x2c36, 0x2416, 0x1bf6, 0x3456, 0x2c36, 0x3c77, 0xcefd, 0x961b, 0xbebc,
0x2c36, 0x2c36, 0x3456, 0x1bf6, 0x44b7, 0xe77e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xef9e, 0x4cd8, 0x1bf5, 0x3456, 0x2c36,
0x3456, 0xb69c, 0x961a, 0x6539, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x1c16, 0x2c36, 0x2c36, 0x2c77,
0x2c57, 0x2310, 0x0906, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0906, 0x2310, 0x2c57,
0x2c77, 0x2c36, 0x2c36, 0x2416, 0x5d18, 0x7579, 0x1bf5, 0x2c56, 0x2c36, 0x8dfa, 0xbebc, 0x5cf8,
0x2416, 0x3456, 0x1bf5, 0x4cf8, 0xefbf, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xf7df, 0x6d59, 0x13d5, 0x3456,
0x2416, 0x3456, 0x6d59, 0x1c16, 0x2c36, 0x2c36, 0x2416, 0x3c77, 0xa63b, 0x2c36, 0x2c36, 0x2c36,
0x3436, 0x3477, 0x2c57, 0x2330, 0x0988, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0988, 0x2330, 0x2c57, 0x2c77, 0x2c36,
0x2c16, 0x2c36, 0x2c36, 0x2416, 0x4cd8, 0xefbe, 0x3c97, 0x2416, 0x2c36, 0x1bf5, 0x1c16, 0x2416,
0x3456, 0x13d5, 0x6d59, 0xf7df, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x8dda, 0x13f5,
0x2c36, 0x2c36, 0x1bf5, 0x2c36, 0x2c36, 0x3c97, 0x1c16, 0xb6bc, 0xb69c, 0x1bf5, 0x3456, 0x2416,
0x13d5, 0x2415, 0x2c36, 0x2c77, 0x2c77, 0x23b3, 0x122b, 0x08c4, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x08c4, 0x122b, 0x23b3, 0x2c77, 0x2c77, 0x2c36, 0x2c36, 0x1c16,
0x3477, 0x3477, 0x2c36, 0x3456, 0x0bb5, 0x8dfa, 0xdf5e, 0x2416, 0x2c36, 0x2c36, 0x3456, 0x2c36,
0x13f5, 0x8dda, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xb6bc,
0x2c36, 0x1c16, 0x3456, 0x3456, 0x1bf6, 0x7579, 0xe77e, 0xe77e, 0x2416, 0x2c36, 0x2416, 0x4497,
0x8e1b, 0x4cd7, 0x1bf6, 0x2c36, 0x2c36, 0x2c57, 0x3498, 0x2c36, 0x2372, 0x122b, 0x0905, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0905, 0x120b, 0x2372, 0x2c56, 0x3498, 0x2c57, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0x6d59,
0xb69c, 0x3c97, 0x2416, 0x3456, 0x2416, 0xae9c, 0xae9c, 0x3456, 0x2c36, 0x3456, 0x1bf6, 0x2c36,
0xb6bc, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xdf5e, 0x54f8, 0x13d5, 0x2c36, 0x3456, 0x1bf6, 0x5d18, 0x5d18, 0x1bf6, 0x3456, 0x1c16, 0xbedc,
0xbebc, 0xcf1d, 0x7dba, 0x1bf6, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c57, 0x2c98, 0x2c56, 0x23b4,
0x1aef, 0x11c9, 0x0946, 0x00a3, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0083, 0x0926, 0x11c9, 0x22ef,
0x23b4, 0x2c57, 0x3498, 0x2c77, 0x2c36, 0x2c36, 0x2c36, 0x1c16, 0x2c36, 0x2c36, 0x1c16, 0xdf5e,
0x9e1b, 0x6d79, 0x3456, 0x2c36, 0x2c56, 0x6538, 0x1bf5, 0x2c36, 0x2c36, 0x13d5, 0x54f8, 0xdf5e,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x961b, 0x2436, 0x1bf6, 0x3456, 0x2416, 0x2416, 0x3456, 0x2416, 0x3c77, 0xf7bf,
0x6d79, 0x2c77, 0xefbf, 0x3456, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c57,
0x2c77, 0x3498, 0x2c77, 0x2c15, 0x23b3, 0x2351, 0x1acf, 0x1a6d, 0x120a, 0x1188, 0x0947, 0x0947,
0x0947, 0x0947, 0x1188, 0x120a, 0x124c, 0x1acf, 0x2351, 0x2393, 0x2c15, 0x2c57, 0x1c57, 0x2c77,
0x2c57, 0x2c36, 0x2415, 0x2c16, 0x2c36, 0x2c36, 0x2c36, 0x9e1b, 0x44b7, 0x2416, 0x2c16, 0x5d18,
0x85da, 0xffdf, 0x6539, 0x1bf5, 0x2c36, 0x2416, 0x3456, 0x2416, 0x2436, 0x961b, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xd73d, 0x54f8, 0x1bf5, 0x2c36, 0x3456, 0x2c36, 0x2c36, 0x2c56, 0x3477,
0x6538, 0x2c77, 0xefbf, 0x3456, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x2c36, 0x2c36, 0x2c36, 0x2c56, 0x2c57, 0x2c77, 0x2c77, 0x3498, 0x3498, 0x3498, 0x2c98, 0x2c78,
0x2c98, 0x2c78, 0x3498, 0x3498, 0x2c78, 0x2c77, 0x2c77, 0x2c57, 0x2436, 0x3c97, 0xae7b, 0x4497,
0x2416, 0x2c36, 0x44b7, 0x3477, 0x2416, 0x3456, 0x13f5, 0xc6fd, 0x8dfa, 0x13d5, 0x3456, 0x13d5,
0x7dba, 0x961b, 0x2c36, 0x2c36, 0x3456, 0x2c36, 0x1bf5, 0x54f8, 0xd73d, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xa67b, 0x3477, 0x1bf5, 0x2c36, 0x2c36, 0xb6bc, 0x8dfa,
0x3cb7, 0xdf5e, 0xa65b, 0x1bf5, 0x2c56, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2415, 0x23f5, 0x2c36, 0x2c36, 0x2c36,
0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x3436, 0x13d5, 0xb6bc, 0x961b, 0x2c36,
0x2c36, 0x2436, 0xbedc, 0xd71d, 0x2c36, 0x2c36, 0x2416, 0x54f8, 0xe77e, 0x2c36, 0x2c36, 0x2c36,
0x3c76, 0x2416, 0x2c36, 0x2c36, 0x1bf5, 0x3477, 0xa67b, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xef9e, 0x85da, 0x2c56, 0x13d5, 0x3c77, 0x9e3b,
0xcf1d, 0x8dfa, 0x2436, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x2416, 0x4cb7, 0xb69c, 0x3456, 0x2c36, 0x2c36, 0x2416, 0x44b7, 0x4cd8, 0x2416, 0x2c36, 0x2c36,
0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0xe77e, 0x7599, 0x13d5,
0x3456, 0x2c36, 0x13f5, 0xcf3d, 0x5d18, 0x1bf5, 0x2c36, 0x2436, 0x9e3b, 0x4497, 0x2416, 0x2c36,
0x2c36, 0x2c36, 0x1bf5, 0x2c56, 0x85da, 0xef9e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xe77e, 0x85da, 0x2436, 0x0395,
0x13d5, 0x1bf6, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x3456,
0x13d5, 0x5d38, 0xd71d, 0x1bf6, 0x3456, 0x1bf5, 0x7dba, 0xcf1d, 0xc6fd, 0x9e1b, 0x23f6, 0x2c36,
0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2416, 0x44b7, 0xdf5d, 0xd71d, 0x3c97,
0x2416, 0x2c36, 0x2c36, 0xcefd, 0xd73d, 0x3477, 0x2c36, 0x2416, 0x54f8, 0xa63b, 0x2416, 0x2c36,
0x1bf5, 0x2c56, 0x7dba, 0xe77e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x2c36, 0xffff, 0xffff, 0xffff, 0x2c36, 0xffff, 0xffff, 0xe77e, 0x961b,
0x3c97, 0x1bf5, 0x1bf6, 0x2c36, 0x2c56, 0x2c56, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2436,
0x3477, 0xae9c, 0xb6bc, 0x13f5, 0x3456, 0x2416, 0xdf3d, 0x44d7, 0x1bf6, 0xef7e, 0x3456, 0x2c36,
0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c56, 0x1bf5, 0x6559, 0xae7c, 0x7599, 0xd71d,
0x1c16, 0x2c36, 0x3477, 0x9e5b, 0xcf1d, 0x2416, 0x3456, 0x2c36, 0x3c97, 0x3c77, 0x13d5, 0x3c97,
0x8e1a, 0xe77e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xf7df, 0xae7c, 0x5d18, 0x2416, 0x1bf5, 0x2416, 0x2c36, 0x3456, 0x3456, 0x2c36, 0x2c36, 0x2416,
0x3477, 0xe77e, 0xc6dc, 0x3456, 0x2c36, 0x1bf5, 0xa65b, 0xae9c, 0xa65b, 0xbebc, 0x2416, 0x2c36,
0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x1bf5, 0x7d99, 0x8dda, 0x1c16, 0x7dba,
0x3456, 0x2c36, 0x2436, 0x2c36, 0x3c97, 0x2c36, 0x2416, 0x1bf5, 0x2416, 0x54f8, 0xae7c, 0xf7df,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xdf5e, 0x961a, 0x54f8, 0x2416, 0x1bf5, 0x1bf5, 0x2436, 0x2c36, 0x2c36,
0x95fa, 0xe77e, 0x4497, 0x2416, 0x2c36, 0x2c36, 0x2c36, 0x85da, 0x8dfa, 0x3456, 0x2c36, 0x2c36,
0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x1bf6,
0x2c56, 0x3456, 0x2c36, 0x1bf6, 0x13d5, 0x2416, 0x54f8, 0x8dfa, 0xdf5e, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xdf5e, 0x9e3b, 0x6559, 0x3c97, 0x1c16, 0x13f5,
0x3c97, 0x3477, 0x2416, 0x2c36, 0x3456, 0x3456, 0x2c36, 0x1bf5, 0x13f5, 0x2c36, 0x2c36, 0x2c36,
0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c36, 0x2c56, 0x3456, 0x3456, 0x2c36, 0x2436, 0x2416, 0x2416,
0x1bf6, 0x1c16, 0x3c97, 0x5d38, 0x9e3b, 0xdf5e, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xefbf, 0xcf3d, 0xa65b,
0x6538, 0x44b7, 0x3c77, 0x2416, 0x13f5, 0x1bf5, 0x1bf6, 0x2436, 0x2c36, 0x2c36, 0x2c36, 0x2c36,
0x2c36, 0x2c36, 0x2c36, 0x2436, 0x2416, 0x1bf6, 0x1bf5, 0x13d5, 0x2416, 0x3c77, 0x44d8, 0x6d59,
0xa65b, 0xcf3d, 0xefbe, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xf7bf, 0xe77e, 0xc6fd, 0xa65b, 0x95fa, 0x7559, 0x5cf8, 0x4477, 0x3457, 0x2c36,
0x2c36, 0x3456, 0x4497, 0x5cf8, 0x7559, 0x95fa, 0xa65b, 0xc6fd, 0xe77e, 0xf7df, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff};

```

3.36 images.h

```
// images.h

#ifndef IMAGES_H
#define IMAGES_H

#include "automatic_icon.h"
#include "brightness_icon.h"
#include "color_icon.h"
#include "empty_sun.h"
#include "ethernet_active.h"
#include "ethernet_icon.h"
#include "font_icon.h"
#include "full_sun.h"
#include "granasat_logo.h"
#include "manual_icon.h"
#include "settings_icon.h"
#include "step_icon.h"

#endif // IMAGES_H
```

3.37 manual_icon.h

```
#if defined(__AVR__)
    #include <avr/pgmspace.h>
#elif defined(__PIC32MX__)
    #define PROGMEM
#elif defined(__arm__)
    #define PROGMEM
#endif

const unsigned short manual_icon[ ] PROGMEM={0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2104, 0xbdf7, 0x2945, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x10a2, 0x0000, 0x0000, 0x0020, 0x0000, 0x630c, 0xffff, 0xbdf7, 0xffff, 0x6b6d, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1082,
0xa514, 0xf79e, 0x5acb, 0x0000, 0x0000, 0x10a2, 0xef5d, 0x8430, 0x0000, 0x7bcf, 0xf79e, 0x18c3,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x39e7, 0xef7d,
0xffff, 0xffff, 0xffff, 0xbdf7, 0x0841, 0x1082, 0xdefb, 0xa534, 0x0020, 0x9cd3, 0xe73c, 0x10a2,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2104, 0xf79e, 0xffff,
0x6b6d, 0x5aeb, 0x9cd3, 0xffff, 0xbdd7, 0x0000, 0x4a49, 0xf79e, 0xe71c, 0xf79e, 0x52aa, 0x0000,
0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x9cd3, 0xffff, 0x4a69,
0x0000, 0x0000, 0x0000, 0xa534, 0xffff, 0x4208, 0x0000, 0x0841, 0x8c51, 0x0861, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xd69a, 0xffff, 0x18c3,
0x0020, 0x0861, 0x0000, 0x738e, 0xffff, 0x6b6d, 0x0000, 0x0000, 0x31a6, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6b4d, 0xffff, 0x39e7,
0x0000, 0x0000, 0x0000, 0xa514, 0xffff, 0x18c3, 0x0861, 0x94b2, 0xd69a, 0x9cf3, 0x0861, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0841, 0xbdf7, 0xffff,
0x5acb, 0x4228, 0x8c71, 0xffff, 0x738e, 0x0000, 0x39e7, 0xe73c, 0x0000, 0xdedb, 0x4a49, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x94b2,
0xffff, 0xffff, 0xe71c, 0x630c, 0x0000, 0x0000, 0x0000, 0x8c71, 0xd69a, 0x94b2, 0x0861, 0x0020,
0x0020, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6b4d, 0xbdf7,
0xef7d, 0xffff, 0xdefb, 0xc618, 0xef5d, 0xd6ba, 0x2104, 0x0000, 0x39e7, 0x0020, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x8c71, 0xdedb, 0xd6ba, 0xdedb, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x9cd3, 0x9cd3, 0x4208, 0x0020, 0x0000, 0x0000, 0x0000, 0x4a49,
0x6b6d, 0x0861, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0x0000, 0x0000, 0x0020, 0x0000, 0x0000, 0xad55, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xef7d, 0x52aa, 0x4208, 0x0000, 0x0000, 0x0000, 0x4a49, 0xce79, 0xffff,
0xbdd7, 0x1082, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xa534, 0xffff, 0xffdf, 0xffdf, 0xffdf, 0xffdf,
0xffdf, 0xffdf, 0xffdf, 0xffdf, 0xffff, 0xffff, 0x9cd3, 0x6b4d, 0xce59, 0xffff, 0xe71c, 0x632c,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xa534, 0xffdf, 0xf7be, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xef7d, 0x632c, 0x0020, 0x0000,
0x0020, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xa514, 0xffff, 0xffff, 0xd6ba, 0x7bef, 0x7bef,
0x7bef, 0x7bef, 0x7bef, 0x7bef, 0x7bef, 0x7bef, 0x73ae, 0x7bcf, 0x2104, 0x0000, 0x0000, 0x0020,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x18c3, 0x73ae, 0x73ae, 0x18c3, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0020,
0x0020, 0x0020, 0x0020, 0x0020, 0x0020, 0x0020, 0x0020, 0x0020, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0020, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff};
```

3.38 settings_icon.h

```
#if defined(__AVR__)
    #include <avr/pgmspace.h>
#elif defined(__PIC32MX__)
    #define PROGMEM
#elif defined(__arm__)
    #define PROGMEM
#endif

```

```
const unsigned short settings_icon[ ] PROGMEM={0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
      0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
      0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
      0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000,
      0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
      0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
      0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
      0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
      0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
      0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
      0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
      0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
      0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
      0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
      0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000,
      0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
      0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000,
      0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
      0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff};
```

3.39 step_icon.h

```
#if defined(__AVR__)
    #include <avr/pgmspace.h>
#elif defined(__PIC32MX__)
    #define PROGMEM
#elif defined(__arm__)
    #define PROGMEM
#endif

const unsigned short step_icon[ ] PROGMEM={0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0x0000, 0xffff, 0xffff, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0x0000, 0x0000, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff,
0xffff, 0xffff, 0xffff, 0xffff, 0xffff, 0xffff};
```

3.40 libraries.h File Reference

This header file groups all the libraries and external dependencies required for the project. Including this file in the code ensures the correct inclusion of all necessary libraries, simplifying dependency management and keeping the main code clean and organized.

```
#include <Arduino.h>
#include <freertos/FreeRTOS.h>
#include <freertos/task.h>
#include <BfButton.h>
#include <TFT_eSPI.h>
#include <EthernetENC.h>
#include <SPI.h>
#include <ThreeWire.h>
#include <RtcDS1302.h>
#include <AccelStepper.h>
#include "icons/images.h"
#include <custom_fonts/custom_fonts.h>
```

3.40.1 Detailed Description

This header file groups all the libraries and external dependencies required for the project. Including this file in the code ensures the correct inclusion of all necessary libraries, simplifying dependency management and keeping the main code clean and organized.

Author

Juan Alberto Serrano Redondo

Included libraries:

- Arduino.h: Provides essential functions for interacting with Arduino hardware.
- FreeRTOS: Enables the use of the real-time operating system (RTOS) on compatible microcontrollers.
- BfButton: Provides advanced functionalities for handling buttons.

- TFT_eSPI: Facilitates communication and control of TFT displays.
- EthernetENC: Provides support for Ethernet communication using the ENC28J60 chip.
- SPI: Serial Peripheral Interface communication protocol.
- ThreeWire: Library for communication on three-wire buses.
- RtcDS1302: Provides functions for interacting with the DS1302 RTC module.
- AccelStepper: Enables control of stepper motors.
- Icons: Defines all icons used in the project's graphical interface.
- Fonts: Defines the fonts used in the project.

Including this file in any part of the project ensures that all necessary dependencies are available and correctly configured.

3.41 libraries.h

Go to the documentation of this file.

```

#ifndef LIBRARIES_H
#define LIBRARIES_H

#include <Arduino.h>
#include <freertos/FreeRTOS.h>
#include <freertos/task.h>
#include <BfButton.h>
#include <TFT_eSPI.h>
#include <EthernetENC.h>
#include <SPI.h>
#include <ThreeWire.h>
#include <RtcDS1302.h>
#include <AccelStepper.h>
#include "icons/images.h" // Includes all necessary icons
#include <custom_fonts/custom_fonts.h>

#endif // LIBRARIES_H
```

3.42 main.cpp File Reference

Main project file.

```
#include <globals/globals.h>
#include <libraries.h>
#include <setup/setup.h>
```

Functions

- void loop ()

  Initial system configuration and RTOS task setup.

3.42.1 Detailed Description

Main project file.

3.42.2 Function Documentation

3.42.2.1 loop()

```
void loop ()
```

Initial system configuration and RTOS task setup.

The setup function is responsible for initializing the system's peripherals and configuring the RTOS tasks. The use of an RTOS eliminates the need for traditional looping within the main loop function.

3.43 motor/motor.cpp File Reference

Implementation file for motor control functions.

```
#include "motor.h"
#include <globals/globals.h>
#include <libraries.h>
```

Functions

- void moveMotor (int steps)
  Moves the stepper motor a specified number of steps.

3.43.1 Detailed Description

Implementation file for motor control functions.

Author

Juan Alberto Serrano Redondo

This file contains the implementation of functions used to control the stepper motor, including moving the motor a specified number of steps.

3.43.2 Function Documentation

3.43.2.1 moveMotor()

```
void moveMotor (
            int steps)
```

Moves the stepper motor a specified number of steps.

This function controls the stepper motor by setting the direction pin and generating step pulses to move the motor. It updates the motor's position and refreshes the screen if the current menu is MANUAL or AUTO.

Parameters

steps	The number of steps to move the motor. Positive values move the motor in one direction, and negative values move it in the opposite direction.

3.44 motor/motor.h File Reference

Header file for motor control functions.

Functions

- void moveMotor (int steps)

  Moves the stepper motor a specified number of steps.

3.44.1 Detailed Description

Header file for motor control functions.

Author

Juan Alberto Serrano Redondo

This file contains the declarations of functions and variables used for controlling a stepper motor. It defines the necessary pins and functions to move the motor and update the screen based on the motor's position.

3.44.2 Function Documentation

3.44.2.1 moveMotor()

```
void moveMotor (
             int steps)
```

Moves the stepper motor a specified number of steps.

This function controls the stepper motor by setting the direction pin and generating step pulses to move the motor. It updates the motor's position and refreshes the screen if the current menu is MANUAL or AUTO.

Parameters

steps	The number of steps to move the motor. Positive values move the motor in one direction, and negative values move it in the opposite direction.

3.45 motor.h

Go to the documentation of this file.

```

#ifndef MOTOR_H
#define MOTOR_H

void moveMotor(int steps);

#endif // MOTOR_H
```

3.46 project_tasks/projectTasks.cpp File Reference

Header file for system task management.

```
#include "projectTasks.h"
#include <motor/motor.h>
#include <GUI/GUI.h>
#include <button/button.h>
#include <libraries.h>
#include <globals/globals.h>
```

Functions

- void taskShowTime (void ∗pvParameters)

  Task to display the current time on the screen.
- void taskStartScreen (void ∗pvParameters)

  Task to initialize and display the startup screen.
- void taskUpdateScreen (void ∗pvParameters)

  Task to update the screen based on the current menu.
- void taskScreenTimeout (void ∗pvParameters)

  Task to manage screen timeout and automatic brightness adjustment.
- void rotarymotorTask (void ∗pvParameters)

  Task for managing the rotary encoder.
- void motorTask (void ∗pvParameters)

  Task for controlling the motor.
- void taskEthernet (void ∗pvParameters)

  Task for managing Ethernet communication.
- void taskEncoder (void ∗pvParameters)

  Task for handling encoder-related tasks.
- void taskButtonPress (void ∗pvParameters)

  Task for handling button press events.

3.46.1 Detailed Description

Header file for system task management.

This header file contains function for managing system tasks, user interface on the screen, Ethernet communication, stepper motor control, and rotary encoder handling.

Author

Juan Alberto Serrano Redondo.

3.46.2 Function Documentation

3.46.2.1 motorTask()

```
void motorTask (
            void * pvParameters)
```

Task for controlling the motor.

This task adjusts the motor speed based on the rotary encoder's state. It operates only when the menu is set to AUTO.

Parameters

pvParameters	Pointer to task parameters (not used).

3.46.2.2 rotarymotorTask()

```
void rotarymotorTask (
            void * pvParameters)
```

Task for managing the rotary encoder.

This task reads the state of the rotary encoder and updates the position and display accordingly. It only operates when the menu is set to AUTO.

Parameters

pvParameters	Pointer to task parameters (not used).

3.46.2.3 taskButtonPress()

```
void taskButtonPress (
            void * pvParameters)
```

Task for handling button press events.

This task processes button presses, including single, double, and long presses, and executes the corresponding handlers.

Parameters

pvParameters	Pointer to task parameters (not used).

3.46.2.4 taskEncoder()

```
void taskEncoder (
            void * pvParameters)
```

Task for handling encoder-related tasks.

This task processes the rotary encoder input to navigate menus and adjust settings. It updates the display based on the current menu and encoder position.

Parameters

pvParameters	Pointer to task parameters (not used).

3.46.2.5 taskEthernet()

```
void taskEthernet (
            void * pvParameters)
```

Task for managing Ethernet communication.

This task handles Ethernet client connections and commands. It provides a menu for controlling the motor and updates the Ethernet status.

Parameters

pvParameters	Pointer to task parameters (not used).

3.46.2.6 taskScreenTimeout()

```
void taskScreenTimeout (
            void * pvParameters)
```

Task to manage screen timeout and automatic brightness adjustment.

Handles screen timeout and controls brightness and automatic shutdown.

Parameters

pvParameters	Pointer to task parameters (not used).

3.46.2.7 taskShowTime()

```
void taskShowTime (
            void * pvParameters)
```

Task to display the current time on the screen.

This task shows the current time on the screen if the current menu is not the initial screen.

Parameters

pvParameters	Pointer to task parameters (not used).

3.46.2.8 taskStartScreen()

```
void taskStartScreen (
            void * pvParameters)
```

Task to initialize and display the startup screen.

Configures and shows the initial screen, then switches to the main menu.

Parameters

pvParameters	Pointer to task parameters (not used).

3.46.2.9 taskUpdateScreen()

```
void taskUpdateScreen (
             void * pvParameters)
```

Task to update the screen based on the current menu.

Updates the screen according to the current menu and commands received in the queue.

Parameters

pvParameters	Pointer to task parameters (not used).

3.47 project_tasks/projectTasks.h File Reference

Header file for system task management.

```
#include <libraries.h>
```

Functions

- void taskUpdateScreen (void *pvParameters)

    Task to update the screen based on the current menu.
- void taskStartScreen (void *pvParameters)

    Task to initialize and display the startup screen.
- void taskShowTime (void *pvParameters)

    Task to display the current time on the screen.
- void taskScreenTimeout (void *pvParameters)

    Task to manage screen timeout and automatic brightness adjustment.
- void rotarymotorTask (void *pvParameters)

    Task for managing the rotary encoder.
- void motorTask (void *pvParameters)

    Task for controlling the motor.
- void taskEthernet (void *pvParameters)

    Task for managing Ethernet communication.
- void taskEncoder (void *pvParameters)

    Task for handling encoder-related tasks.
- void taskButtonPress (void *pvParameters)

    Task for handling button press events.

3.47.1 Detailed Description

Header file for system task management.

This header file contains function declarations for managing system tasks, user interface on the screen, Ethernet communication, stepper motor control, and rotary encoder handling.

Note

Ensure necessary libraries and definitions are included in the corresponding .cpp file.

Adjust constants and configurations according to hardware requirements.

3.47.2 Function Documentation

3.47.2.1 motorTask()

```
void motorTask (
            void * pvParameters)
```

Task for controlling the motor.

This task adjusts the motor speed based on the rotary encoder's state. It operates only when the menu is set to AUTO.

Parameters

pvParameters	Pointer to task parameters (not used).

3.47.2.2 rotarymotorTask()

```
void rotarymotorTask (
            void * pvParameters)
```

Task for managing the rotary encoder.

This task reads the state of the rotary encoder and updates the position and display accordingly. It only operates when the menu is set to AUTO.

Parameters

pvParameters	Pointer to task parameters (not used).

3.47.2.3 taskButtonPress()

```
void taskButtonPress (
            void * pvParameters)
```

Task for handling button press events.

This task processes button presses, including single, double, and long presses, and executes the corresponding handlers.

Parameters

pvParameters	Pointer to task parameters (not used).

3.47.2.4 taskEncoder()

```
void taskEncoder (
             void * pvParameters)
```

Task for handling encoder-related tasks.

This task processes the rotary encoder input to navigate menus and adjust settings. It updates the display based on the current menu and encoder position.

Parameters

pvParameters	Pointer to task parameters (not used).

3.47.2.5 taskEthernet()

```
void taskEthernet (
             void * pvParameters)
```

Task for managing Ethernet communication.

This task handles Ethernet client connections and commands. It provides a menu for controlling the motor and updates the Ethernet status.

Parameters

pvParameters	Pointer to task parameters (not used).

3.47.2.6 taskScreenTimeout()

```
void taskScreenTimeout (
             void * pvParameters)
```

Task to manage screen timeout and automatic brightness adjustment.

Handles screen timeout and controls brightness and automatic shutdown.

Parameters

pvParameters	Pointer to task parameters (not used).

3.47.2.7 taskShowTime()

```
void taskShowTime (
             void * pvParameters)
```

Task to display the current time on the screen.

This task shows the current time on the screen if the current menu is not the initial screen.

Parameters

pvParameters	Pointer to task parameters (not used).

3.47.2.8 taskStartScreen()

```
void taskStartScreen (
             void * pvParameters)
```

Task to initialize and display the startup screen.

Configures and shows the initial screen, then switches to the main menu.

Parameters

pvParameters	Pointer to task parameters (not used).

3.47.2.9 taskUpdateScreen()

```
void taskUpdateScreen (
             void * pvParameters)
```

Task to update the screen based on the current menu.

Updates the screen according to the current menu and commands received in the queue.

Parameters

pvParameters	Pointer to task parameters (not used).

3.48 projectTasks.h

Go to the documentation of this file.

```

#ifndef PROJECTTASK_H
#define PROJECTTASK_H
#include <libraries.h>

// LCD tasks
void taskUpdateScreen(void *pvParameters);
void taskStartScreen(void *pvParameters);
void taskShowTime(void *pvParameters);
void taskScreenTimeout(void *pvParameters);

// Motor tasks
void rotarymotorTask(void *pvParameters);
void motorTask(void *pvParameters);

// Ethernet tasks
void taskEthernet(void *pvParameters);

// Rotary encoder tasks
void taskEncoder(void *pvParameters);
void taskButtonPress(void *pvParameters);

#endif // PROJECTTASK_H
```

3.49 setup/setup.h File Reference

Contains all setup configurations for the project including Ethernet, LCD, Motor, Pin, RTC, and Task setups.

```
#include <globals/globals.h>
#include <libraries.h>
#include <GUI/GUI.h>
#include <project_tasks/projectTasks.h>
```

Functions

- void setup ()

3.49.1 Detailed Description

Contains all setup configurations for the project including Ethernet, LCD, Motor, Pin, RTC, and Task setups.

3.49.2 Function Documentation

3.49.2.1 setup()

```
void setup ()
```

Ethernet setup configuration for ENC28J60 module.

Initializes SPI communication and configures the Ethernet with the provided MAC address and IP.

< Initializes SPI communication

< CS pin for ENC28J60 module

< Start Ethernet with MAC address and IP

LCD setup configuration for the TFT display.

Configures serial communication, initializes the display, sets text properties, and creates necessary queues and semaphores for screen management.

< Initialize serial communication at 115200 baud

< Initialize TFT display

< Disable text wrapping

< Set text size to small

< Set display rotation

< Create a queue for screen updates

< Set initial screen brightness

< Enable byte swapping for better display rendering

< Create a mutex for controlling screen updates

Motor setup configuration.

Configures motor speed, acceleration, and reads initial state from encoder.

< Read initial state of the encoder

< Set maximum motor speed

< Set motor acceleration

Pin setup configuration.

Configures the pins for the encoder and motor control.

< Set CLK pin as input with pull-up resistor

< Set DT pin as input with pull-up resistor

< Set direction pin as output

< Set step pin as output

RTC setup configuration.

Initializes the Real-Time Clock (RTC) and sets the date and time based on the compilation time. Verifies if the RTC is write-protected and ensures it is running.

< Initialize RTC communication

< Display the compilation date

< Display the compilation time

< Set compiled date and time

Check if the RTC's date and time are valid, and if not, set the compiled date and time.

< Set RTC to compilation time

Check if the RTC is write-protected, and disable write protection if necessary.

< Disable write protection

Check if the RTC is running, and start it if it is not.

< Start RTC

< Get the current date and time from the RTC

Compare the RTC time with the compiled time and update the RTC if necessary.

< Update RTC to compiled date and time

Task setup configuration.

Creates various FreeRTOS tasks for managing display, button presses, Ethernet, motor control, and screen updates.

< Create task for displaying time

< Create task for starting the screen (core 0)

< Create task for reading encoder (core 0)

< Create task for managing screen timeout

< Create task for handling button presses

< Create task for Ethernet communication

< Create task for updating screen (core 1)

< Create task for rotary motor control

< Create task for motor control

< Start the FreeRTOS scheduler

3.50 setup.h

Go to the documentation of this file.

```
#ifndef SETUP_H
#define SETUP_H

#include <globals/globals.h>
#include <libraries.h>
#include <GUI/GUI.h>
#include <project_tasks/projectTasks.h>
void setup(){
    SPI.begin();
    Ethernet.init(26);
    Ethernet.begin(mac, ip);
    Serial.begin(115200);
    tft.init();
    tft.setTextWrap(false);
    tft.setTextSize(1);
    tft.setRotation(1);
    screenUpdateQueue = xQueueCreate(10, sizeof(ScreenUpdateCommand));
    setBrightness(brightnessLevel);
    tft.setSwapBytes(true);
    gatekeeper = xSemaphoreCreateMutex();
    aLastState = digitalRead(CLK);
    motor.setMaxSpeed(2000.0);
    motor.setAcceleration(1000.0);
    pinMode(CLK, INPUT_PULLUP);
    pinMode(DT, INPUT_PULLUP);
    pinMode(DIR_PIN, OUTPUT);
    pinMode(STEP_PIN, OUTPUT);
    Rtc.Begin();
    Serial.print("compiled: ");
    Serial.print(__DATE__);
    Serial.print(" ");
    Serial.println(__TIME__);
    RtcDateTime compiled = RtcDateTime(__DATE__, __TIME__);
    if (!Rtc.IsDateTimeValid()) {
        Serial.println("RTC lost confidence in the DateTime! Setting compilation date and time.");
        Rtc.SetDateTime(compiled);
    }

    if (Rtc.GetIsWriteProtected()) {
        Serial.println("RTC was write protected, enabling writing now");
        Rtc.SetIsWriteProtected(false);
    }

    if (!Rtc.GetIsRunning()) {
        Serial.println("RTC was not actively running, starting now");
        Rtc.SetIsRunning(true);
    }

    RtcDateTime now = Rtc.GetDateTime();
    if (now < compiled) {
        Serial.println("RTC is older than compile time! Updating DateTime.");
        Rtc.SetDateTime(compiled);
    } else if (now > compiled) {
        Serial.println("RTC is newer than compile time. (this is expected)");
    } else if (now == compiled) {
        Serial.println("RTC is the same as compile time! (not expected but all is fine)");
    }

    xTaskCreatePinnedToCore(taskShowTime, "ShowTime", 16384, NULL, 1, NULL, 1);
    xTaskCreatePinnedToCore(taskStartScreen, "TaskScreenStart", 16384, NULL, 5, NULL, 0);
    xTaskCreatePinnedToCore(taskEncoder, "TaskEncoder", 16384, NULL, 3, NULL, 0);
    xTaskCreatePinnedToCore(taskScreenTimeout, "ScreenTimeout", 16384, NULL, 10, NULL, 1);
    xTaskCreatePinnedToCore(taskButtonPress, "ButtonPress", 16384, NULL, 3, NULL, 0);
    xTaskCreatePinnedToCore(taskEthernet, "TaskEthernet", 16384, NULL, 7, NULL, 1);
    xTaskCreatePinnedToCore(taskUpdateScreen, "UpdateScreen", 16384, NULL, 2, NULL, 0);
    xTaskCreate(rotarymotorTask, "Rotary motor Task", 2048, NULL, 4, NULL);
    xTaskCreate(motorTask, "Motor Task", 2048, NULL, 6, NULL);
    vTaskStartScheduler();
    ScreenUpdateCommand initialCommand = UPDATE_MAIN_MENU;
    xQueueSend(screenUpdateQueue, &initialCommand, portMAX_DELAY);
}
#endif // COMBINED_SETUP_H
```

Index