Milkshake Machine

Jesse Jensen
Kylie Williamson

The purpose of this report is to document the workings of the Milkshake Machine. Scope of the project, requirements, software and hardware design are all outlined. Detailed code and hardware schematics are also provided in the Appendixes.

Introduction

The Milkshake Machine is a machine designed to create a perfectly consistent milkshake with little user effort. Following simple steps, anyone can operate a milkshake machine. This machine requires components such as a blender, a DC motor pump, a breadboard, and some electrical components to build.

The Milkshake Machine requires two simple steps in order to be used. First, the user must place one cup (8 oz) of ice cream into the Milkshake Machine. Then, using a touch screen, the user selects the desired thickness for their milkshake from a list of either ‘Thick’, ‘Medium’, or ‘Thin’. The Milkshake Machine then automatically adds milk to the blender and turns the blender on to the correct speed and length to create a milkshake of the consistency chosen.

The Milkshake Machine is built using a combination of several electrical and mechanical components. First of all, this machine uses a Mainstay six speed blender as its source of mixing the milkshakes. The machine also uses a breadboard in order to connect components, a power relay to switch on AC power to the blender, an LCD touch screen, a DC motor pump, and other electrical components. An in-depth list of required hardware as well as instructions on how to assemble will be detailed later in this report.

Scope

This report is designed to outline the requirements, operation, design, and testing of the Milkshake Machine. Requirements meet demands outlined by the users and funders of the Milkshake Machine project. The operation description gives an overview of how the machine operates. This report outlines design details of both hardware and software along with reasons for specific design choices. Testing verifies that the machine is operable and successfully meets requirements outlined by the requirement stakeholders and is described at the end of the report. Figures, schematics, and code are included in the appendix of this document.

The mechanical design for the project is not included in this report.

Design Overview

Requirements

The following are the requirements for the milkshake machine:

1. Take an input from the LCD screen to specify thickness
    a. Three different thickness settings: Thin, Medium, and Thick
2. An automatic milk dispenser will disperse milk into a blender
    a. The amount of milk will vary depending on thickness
3. The blender will blend for an appropriate amount of time
4. The machine must not need to be reset between milkshakes

Dependencies

The following are dependencies for the milkshake machine:

- A 3.3 volt DC power supply
- 16.0 MHz crystal oscillator
- A blender
- A 120 volt AC power supply
- An automatic pump
- A 3 volt DC power supply
- A 5 volt power relay
- Two transistors
- ILI9341 LCD Screen - XPT2046 Touch Screen

Theory of Operation

After powering on, the LCD screen will display three options the user can choose between: thin, medium, and thick. The user then inserts one 8 oz scoop of ice cream into the blender and closes the lid before selecting an option. Once the user selects an option, the automatic pump begins dispensing milk into the blender. Once the dispenser is done, the blender turns on after a few seconds. Once the blender is done, the milkshake is finished and the LCD screen resets.

Design Details

Hardware Design

The Milkshake machine is implemented on a TivaTM TM4C123GH6PM microcontroller. The display is on a ILI9341 LCD display that also functions as a XPT2046 touch screen. The system uses a blender and a pump. A high-level diagram of the product and its connections, excluding transistors and relays, are shown in Figure 1.

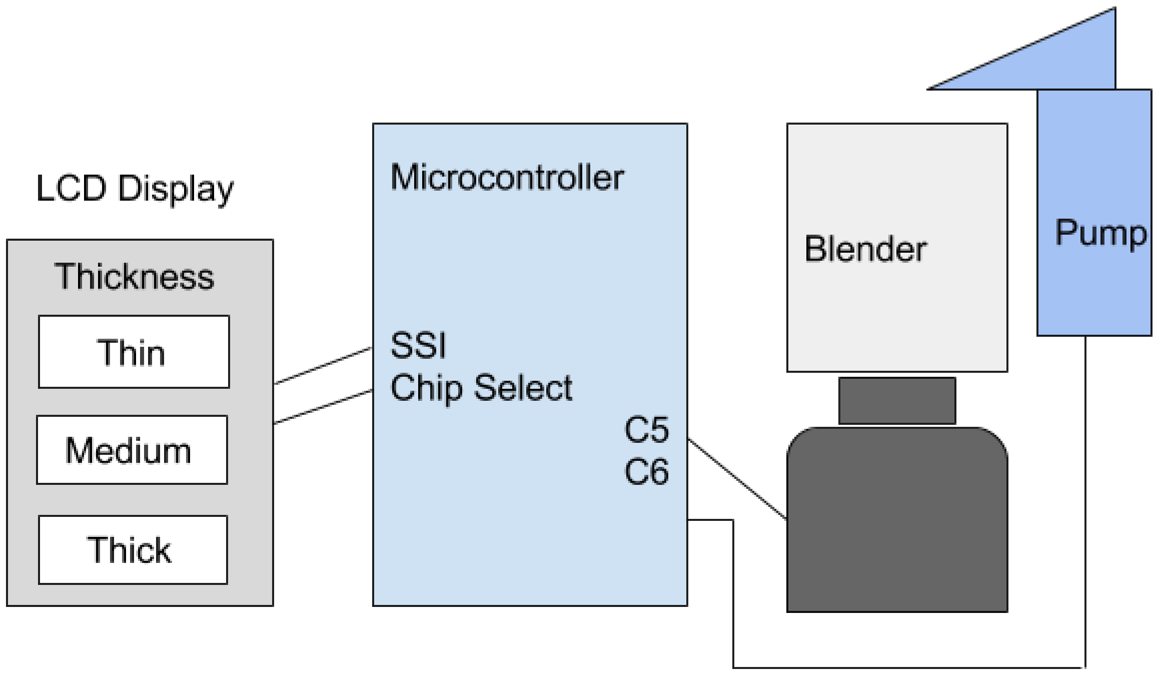


Figure 1 High level hardware layout

The LCD Display is connected to the microcontroller using the SSI. The Microcontroller's Synchronous Serial Interface (SSI) clock, chip select, Rx, and Tx are connected to pins A2-5, respectively. The chip select of the touch screen is connected to pin C4 of the microcontroller, and manually controlled in the program. The LCD screen is powered through the VBUS pin of the microcontroller, and the RESET pin is connected to the 3.3V power.

The blender is connected to the microcontroller through a power relay. The microcontroller amplifies its 3.3 volt signal (pin C5) using a transistor, into a 5 volt signal. That 5 volt signal connects to the power relay, which acts like a switch to turn the blender on.

The pump is connected to a transistor, which acts as a switch. The microcontroller is connected to the base of the transistor and can power the pump through pin C6.

A detailed schematic of all these connections can be found in Appendix A.

Software Design

The software used in this project breaks down into three separate sections: microcontroller initialization, start screen creation, and interrupt handling. The microcontroller initialization section handles all clock, pin, and alternate function initialization necessary for the project. Start screen creation involves LCD initialization and functions for writing commands and data to the touch screen. Interrupt handling includes how the program will react to a user touching the screen. Code for the software described can be seen in Appendix B through Appendix D. An overview of the software can be seen in Figure 2.

Microcontroller initialization. Clock initializations, pin initializations, and SSI initializations must precede programming the microcontroller. First, the clocks for the GPIO ports and the SSI ports must be enabled. The SSI is then initialized by turning on the alternate functions for GPIO pins A2 through A5 (the second alternate function is selected). The aforementioned pins are then enabled. The microcontroller is then specified as master, the system clock is used as the SSI clock source, and a division factor of 0x8 must then be specified. The SSI is declared as 8-bit and, last of all, the SSI is enabled.

Other GPIO pins are then enabled including pins B0, A6, C4, C5, and C6. B0 will act as the command/data pin for LCD/touch screen correspondence and is therefore set as a pull-up output pin. A6 will be the touch screen interrupt pin, so is, therefore, an input pin (default). C4 is the chip select pin for the touch screen and is therefore set to output and enabled. C5 controls the automatic DC pump, and C6 controls the power relay connecting the blender.

Start screen creation. This section uses code provided for the LCD lab (lab 05) for LCD screen initialization, as well as new software to prepare the touch screen for use. The lcd.c and lcd.h files seen in Appendix C contain the code described in this section. The lcd_init function is code provided to us used to setup the LCD screen. The write_dat and write_cmd functions write data and commands (respectively) to the FIFO of the SSI.

A function, write_letter, is created in order to write letters to the LCD screen. This function works by setting each bit in a given byte to either on or off. A background and foreground color are passed in as parameters. The number of bits corresponds to the number of pixels in a 32x24 region. If the pixel for the given region is labeled as ‘on’, the foreground color is displayed. If the pixel is labeled as ‘off’, the background color is displayed. The values for ‘on’ and ‘off’ for each pixel are predefined in an array in Appendix D.

The start screen uses two more functions, write_word and draw_box, to operate. Write_word simply calls the write_letter command to create words of a given size and color. Draw_box defines rows and columns of a box to draw on the LCD screen.

The start_screen function displays a title, “Thickness”. Three boxes are then displayed with their appropriate labels, “Thick”, “Medium”, and “Thin”.

Interrupt handling. Interrupt handling is where most of the functionality behind the hardware of this project takes place, including turning on the DC motor pump, starting timers, and turning on the blender. When the interrupt is triggered, the chip selects of the LCD screen and touch screen must be changed a setting of 'on' for the touch screen and 'off' for the LCD screen. An average coordinate is then calculated from ten input samples from the touch screen. The LCD chip select is then set to 'on' and the touch screen chip select to 'off'.

The function checkBounds checks whether the input given is within the buttons displayed on the touch screen and, if the input is within the specified bounds, determines which button is being pressed. One of three settings is determined based off of the pressed button: thick, medium, or thin. GPIO pin C5 is then sent a signal, which is hooked up to the DC motor pump to begin pumping milk into the blender. A timer is set up using the microcontroller's SysTick peripheral to run for one second. This calculation is computed using the system clock. The equation is given by

$$time = (RELOAD + x) * \left(\frac{1}{coreclock}\right)$$

Where x is the number of instructions additional to the count the timer executes (in this case, 5). The RELOAD value entered is 0xF423FB (15,999,995 decimal), which was calculated to be one second. According to which thickness the user selected, this function is called either 15, 20, or 25 times to instruct the milk pump on how long to operate. The milk pump signal is then turned off and a five second delay is given using the timer described above. A signal is then sent to the power relay allowing an AC current connection from an outlet to the blender to be established and turning the blender on. The blender is timed using the same SysTick timer previously described to blend for 10 seconds. The LCD screen is then re-printed to its original start screen and the touch screen interrupt is re-enabled.

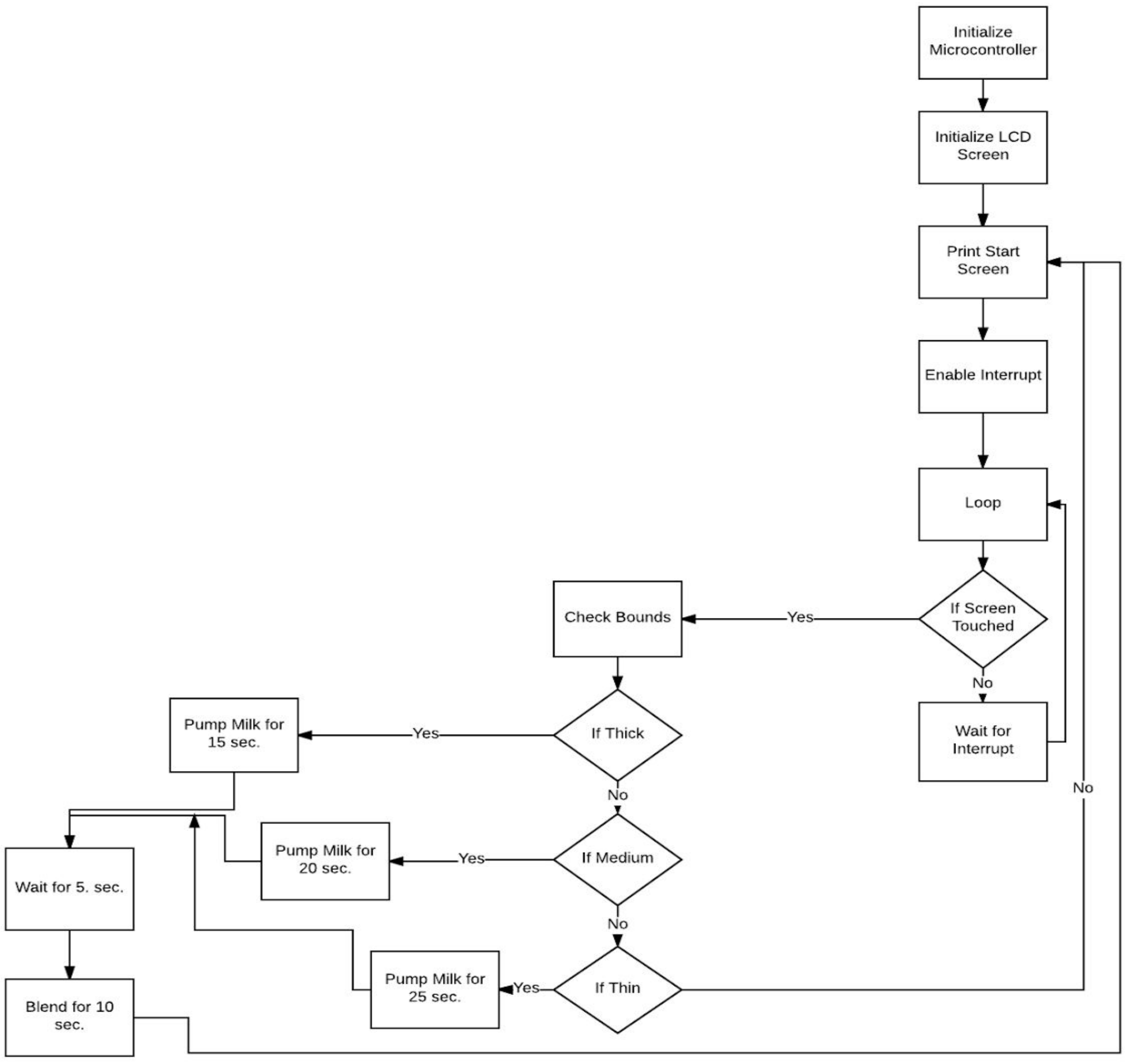


Figure 2. Overview of code

Testing and Verifications

All the requirements were tested as outlined below.

Test 1 - Blender Hardware

The first test verifies that the blender is working correctly. The blender is connected to the power relay, and 5V power is supplied to the relay. The blender is also connected to an 120 volt outlet. Verify that the blender turns on. These tests were performed, and the hardware passed. This test verified part of requirement 3.

Test 2 - Pump Hardware

The second test verifies that the pump is working correctly. The pump is connected to the microcontroller through a transistor, and 3V power is supplied to the pump, and through the GPIO pin. Verify that the pump turns on, and is supplied with enough power to pump liquid. These tests were performed, and the hardware passed. This test verified part of requirement 2.

Test 3 - Timer Rate

The third test verifies that the timer is working properly. A GPIO pin is controlled by the timer to turn on and off when the timer starts and stops. As our program uses one-second timers,that value is tested for one second runs, as it should. An oscilloscope is connected to the GPIO pin, and the timing is tested. This test was successfully performed, as shown in Figure 1. This test helps verify part of requirement 2.

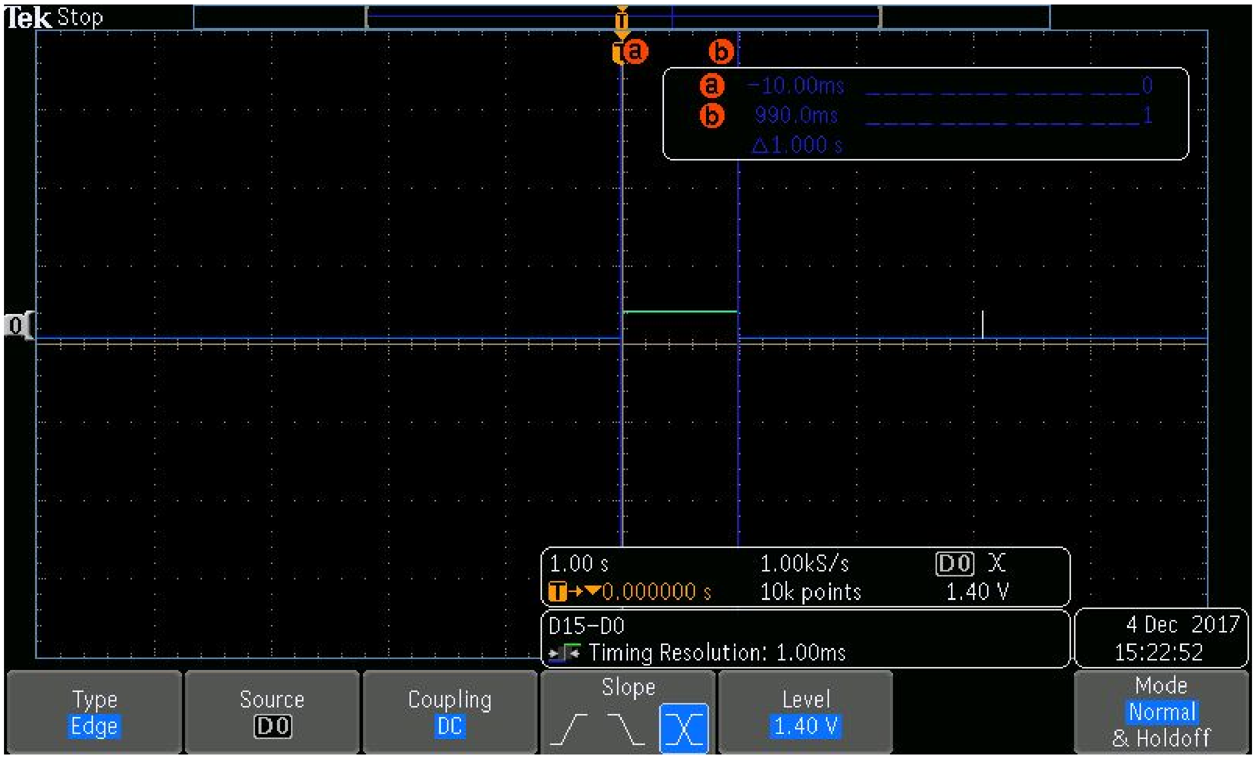


Figure 3. One second timer verification

Test 4 - LCD Input

This test verifies through observation that the LCD input works correctly to meet user requirement 1. That is, that the user can select their desired thickness using the LCD touch screen. This test passes when the LCD screen successfully displays the appropriate text given, three distinct boxes of appropriate size, and consistent colors. The screen must reload after every cycle and allow for the user to select another milkshake. These tests were successfully verified.

Test 5 - Different Thickness Settings

This test verifies that the machine produces milkshakes of appropriate thickness for the user. The milkshake is tested with one serving of ice cream (8 oz). The desired thickness is selected on the touch screen and the cycle completes, blending a milkshake. The test is verified through observation if the consistency of the milkshake is different with each milkshake setting. This test was performed over 10 times with variable thicknesses selected with successful results. This test verifies requirement 1a.

Test 6 - Multiple Run Test

This test verifies that the machine can make multiple milkshakes consecutively without needing to be reset. The machine is prepared with 8 oz of ice cream and placed through a cycle, then prepared and cycled through again. This test passes if multiple cycles are successfully completed without resetting or unplugging the system. This test was completed with successful results with multiple cases. This test verifies requirement 4.

Test 7 - LCD Screen Button Boundaries

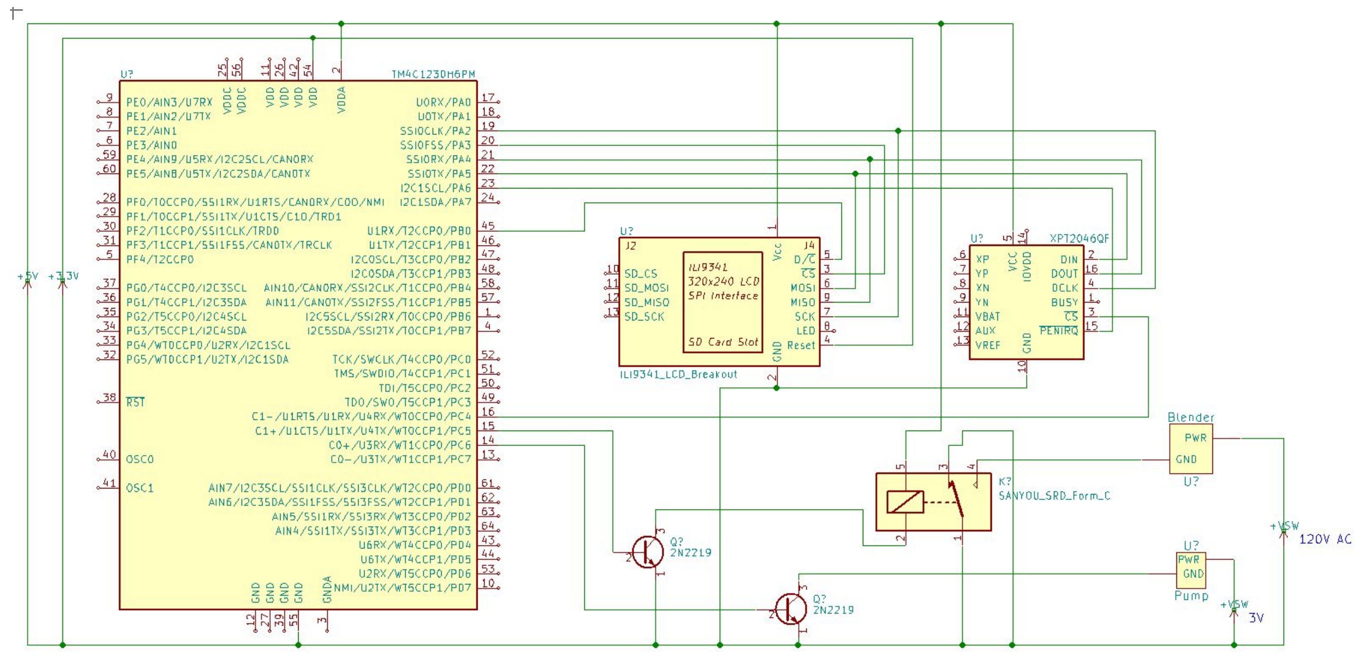
This test ensures that the boundaries on the LCD screen buttons are functioning correctly. The screen is loaded, and each button is tested on whether or not a interrupt is triggered when pressed in specific locations. The interrupt should trigger when the center and corners of the colored boxes are touched. The interrupt should not trigger when the screen is touched outside of those corners. These tests were performed successfully. This test verifies part of requirement 1.

Conclusion

The design described in this document provides a functional prototype of the Milkshake Making Machine. However, the implementation is not yet optimal.

One weakness the code has is that the timer function can’t process a value much greater than a wait of one second, because of data type overflow issues. A more optimal solution would be some sort of function that waits for some number of seconds, to make the code more readable.

Appendix - Hardware Schematic



Appendix B - Main.c

```
#include "lcd.h"

volatile unsigned char* RCGC = (unsigned char*) 0x400FE000;
volatile unsigned int* PortC = (unsigned int*) 0x40006000;
volatile unsigned char* PortB = (unsigned char*) 0x40005000;
volatile unsigned int* SSI = (unsigned int*) 0x40008000;
volatile unsigned int* peripherals = (unsigned int*) 0xe000e000;
volatile unsigned int* PortA = (unsigned int*) 0x40004000;
volatile unsigned int one_sec = 0xF423FB;

volatile char thick = 0;
volatile char medium = 0;
volatile char thin = 0;

//enables necessary interrupts for touch screen
void enable_interrupt(void){
	///////////Set PA6 as interrupt pin/////////////
	//TODO add 2 timer interrupts

	//enable interrupts on port A
	peripherals[0x100/4] |= 0x1;

	//don't read on both edges - IBE
	PortA[0x408/4] = 0x0;

	//configure clock interrupt on edge - IS
	PortA[0x404/4] = 0x0;

	//interupt on falling edge - IEV
	PortA[0x40C/4] = 0x0;

	//set pin as interrupt - IM
	PortA[0x410/4] = 0x40;

}

//Returns the coordinates for the touchscreen
short GetCoord(char cmd){
	int RxNE;
	int Busy;
	int trash;
	int data1;
	int data2;

	Busy = (SSI[0x0C/4] >> 4) & 0x1;
	RxNE = (SSI[0x0C/4] >> 2) & 0x1;
	while(Busy == 0x01){
		Busy = (SSI[0x0C/4] >> 4) & 0x1;
	}
	if(RxNE){
		trash = SSI[0x08/4];
	}

	write_cmd(cmd);
	Busy = (SSI[0x0C/4] >> 4) & 0x1;
	RxNE = (SSI[0x0C/4] >> 2) & 0x1;
	while(RxNE ==  0x00 || Busy == 0x01){
		Busy = (SSI[0x0C/4] >> 4) & 0x1;
		RxNE = (SSI[0x0C/4] >> 2) & 0x1;
	}
	trash = SSI[0x08/4];

	write_dat(0x00);
	Busy = (SSI[0x0C/4] >> 4) & 0x1;
	RxNE = (SSI[0x0C/4] >> 2) & 0x1;
	while(RxNE ==  0x00 || Busy == 0x01){
		Busy = (SSI[0x0C/4] >> 4) & 0x1;
		RxNE = (SSI[0x0C/4] >> 2) & 0x1;
	}
	data1 = SSI[0x08/4];

	write_dat(0x00);
	Busy = (SSI[0x0C/4] >> 4) & 0x1;
	RxNE = (SSI[0x0C/4] >> 2) & 0x1;
	while(RxNE ==  0x00 || Busy == 0x01){
		Busy = (SSI[0x0C/4] >> 4) & 0x1;
		RxNE = (SSI[0x0C/4] >> 2) & 0x1;
	}
	data2 = SSI[0x08/4];

	return ((data1 << 8) | data2) >> 3;
}

//initialize ports for blender, pump, and touchscreen
void initialize(void)
{
	RCGC[0x608] = 0x7;	//RCGCGPIO
	RCGC[0x61c] = 0x1;	//RCGCSSI

	//B0 is output
	PortB[0x400] = 0x1;
	//B0 is pull-up
	PortB[0x510] = 0x1;
	//Enable B0
	PortB[0x51c] = 0x1;


	////////SSIO initialize///////////
	//A5:2 is output
	PortA[0x400/4] = 0x3c;
	//Alternate funtions for A5:2 - clock
	PortA[0x420/4] = 0x3c;
	//we need alternate function 2
	PortA[0x52c/4] = 0x00222200;
	//Enable A6:2
	PortA[0x51c/4] = 0x7c;


	//Chip select
	//C4 is output
	PortC[0x400/4] = 0x70;
	//enable C4
	PortC[0x51c/4] = 0x70;
	PortC[0x3fc/4] = 0x10;

	//Clear SSE Bit
	SSI[0x4/4] &= 0x00;	//master
	//Clock is system clock
	SSI[0xFC8/4] = 0x0;
	//Division factor (change) 8MHz
	SSI[0x10/4] = 0x8;	//PRE


	SSI[0x0] = 0xC7;		//8 bit
	//enable SSI
	SSI[0x4/4] |= 0x2;

}

//runs timer for time amount of time
void run_timer(int time){

	char done = 0;

	peripherals[0x10/4] &= ~(0x8000);		//sets COUNT bit to zero
	peripherals[0x14/4] = time;
	peripherals[0x18/4] = time;
	peripherals[0x10/4] |= 0x5;	//enable SysTick and use System Clock

	while(!done){
		done = (peripherals[0x10/4] >> 16) & 1;
	}
}

// check if touch was in the buttons we defined
void checkBounds(short x, short y)
{
	if( 0x2E3 < x && x < 0x950 && 0x2C2 < y && y < 0xB71 )
	{
		if (0x900 < y) //////checks if in yellow box
		{
				draw_box(30, 248, 170, 50, 0x69C9);
				write_word(30, 253, "Thin", 4, 0xFFFF, 0x69C9);
				thin = 1;
		}

		else if(0x580 < y && y < 0x870) //checks if in green
		{
				draw_box(30, 145, 170, 50, 0x6ACB);
```

```
                        write_word(30, 150, "Medium", 6, 0xFFFF, 0x6ACB);
                        medium = 1;
                }

                else if(y < 0x64E) //checks if in red
                {
                        draw_box(30, 52, 170, 50, 0x6ACF);
                        write_word(30, 57, "Thick", 5, 0xFFFF, 0x6ACF);
                        thick = 1;
                }

        }

}

//default screen
void start_screen(void){
        draw_box(0, 0, 240, 320, 0x0001);
        write_word(0,0,"Thickness",9, 0xFFFF, 0x0001);

        draw_box(20, 42, 190, 70, 0x6ACF);
        draw_box(20, 135, 190, 70, 0x6ACB);
        draw_box(20, 238, 190, 70, 0x69C9);

        draw_box(30, 52, 170, 50, 0x0001);
        write_word(30, 57, "Thick", 5, 0xFFFF, 0x0001);
        draw_box(30, 145, 170, 50, 0x0001);
        write_word(30, 150, "Medium", 6, 0xFFFF, 0x0001);
        draw_box(30, 248, 170, 50, 0x0001);
        write_word(30, 253, "Thin", 4, 0xFFFF, 0x0001);
}

//function that runs when the touchscreen is touched
void GPIOA_Handler(void){
        int pumpTimer;
        int i = 0;
        short xCoord = 0;
        short yCoord = 0;

        //set SSI0 LCD chip select to off (high)
        PortA[0x3fc/4] = 0x04;

        //set SSI0 Touch Screen select to on (low)
        PortC[0x3fc/4] &= ~(0x0010);

        for (; i < 10; i++){
                xCoord += GetCoord(0xD0);               //X-coord
                yCoord += GetCoord(0x90);               //Y-coord
        }
        xCoord /= 10;
        yCoord /= 10;

        //LCD to on
        PortA[0x3fc/4] &= 0x0000;
```

```
//Touch Screen to Off
PortC[0x3fc/4] |= 0x0010;

checkBounds(xCoord, yCoord);

//timing for thick milkshakes
if (thick)
{
        thick = 0;

        PortC[0x3fc/4] |= 0x20;
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        PortC[0x3fc/4] &= ~(0x20);

        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);

        PortC[0x3fc/4] |= 0x40;
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        run_timer(one_sec);
        PortC[0x3fc/4] &= ~(0x40);
}
//timing for medium milkshakes
else if (medium)
{
        medium = 0;

        PortC[0x3fc/4] |= 0x20;
        run_timer(one_sec);
```

```
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                PortC[0x3fc/4] &= ~(0x20);

                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);

                PortC[0x3fc/4] |= 0x40;
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                PortC[0x3fc/4] &= ~(0x40);
        }
        //timing for thin milkshakes
        else if(thin)
        {
                thin = 0;

                PortC[0x3fc/4] |= 0x20;
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                PortC[0x3fc/4] &= ~(0x20);

                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);

                PortC[0x3fc/4] |= 0x40;
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                run_timer(one_sec);
                PortC[0x3fc/4] &= ~(0x40);
        }

        start_screen();

        //re-enable touch-screen
        PortA[0x41c/4] = 0x40;
}


int main(void)
{
        //your code here.
        initialize();

        lcd_init();
        start_screen();

        enable_interrupt();

        while(1);
}
```

Appendix C - lcd.h

```
#ifndef LCD_C
#define LCD_C

//definitions of useful colors.
// page 238 of the data sheet.
#define white        0xFFFF
#define black        0x0001
#define grey         0xF7DE
#define blue         0x001F
#define red          0xF800
#define magenta      0xF81F
#define green        0x07E0
#define cyan         0x7FFF
#define yellow       0xFFE0

//This function configures the LCD screen for using.
//It assumes that there exist write_dat and write_cmd functions that function properly and that peripherals
are configured and enabled.
void lcd_init(void);

//This function writes data to the LCD screen
//It assumes that peripherals are configured configured and enabled
void write_dat(unsigned char data);

//This function writes a command to the LCD screen
//It assumes that peripherals are configured configured and enabled
void write_cmd(unsigned char command);

void write_dat_2(short data);

void draw_box(short startX, short startY, short width, short height, short color);

void write_word(short startX, short startY, char word[], int size, short color, short backColor);
#endif
```

Appendix C - lcd.c

```
#include "lcd.h"
#include "Ubuntu-1.h"

unsigned int* B0 = (unsigned int*) 0x40005000;
unsigned int* SSI0 = (unsigned int*) 0x40008000;


//This function configures the LCD screen for using.
//It assumes that there exist write_dat and write_cmd functions that function properly and that peripherals
are configured and enabled.
void lcd_init(void)
{
   int i;
   write_cmd(0xCB);
   write_dat(0x39);
   write_dat(0x2C);
   write_dat(0x00);
   write_dat(0x34);
   write_dat(0x02);

   write_cmd(0xCF);
   write_dat(0x00);
   write_dat(0XC1);
   write_dat(0X30);

   write_cmd(0xE8);
   write_dat(0x85);
   write_dat(0x00);
   write_dat(0x78);

   write_cmd(0xEA);
   write_dat(0x00);
   write_dat(0x00);

   write_cmd(0xED);
   write_dat(0x64);
   write_dat(0x03);
   write_dat(0X12);
   write_dat(0X81);

   write_cmd(0xF7);
   write_dat(0x20);

   write_cmd(0xC0);    //Power control
   write_dat(0x23);   //VRH[5:0]

   write_cmd(0xC1);    //Power control
   write_dat(0x10);   //SAP[2:0];BT[3:0]

   write_cmd(0xC5);    //VCM control
   write_dat(0x3e);
   write_dat(0x28);

   write_cmd(0xC7);    //VCM control2
   write_dat(0x86);

   write_cmd(0x36);    // Memory Access Control
   write_dat(0x48);        //C8

   write_cmd(0x3A);
   write_dat(0x55);

   write_cmd(0xB1);
   write_dat(0x00);
   write_dat(0x18);

   write_cmd(0xB6);    // Display Function Control
   write_dat(0x08);
   write_dat(0x82);
   write_dat(0x27);

   write_cmd(0xF2);    // 3Gamma Function Disable
   write_dat(0x00);

   write_cmd(0x26);    //Gamma curve selected
   write_dat(0x01);

   write_cmd(0xE0);    //Set Gamma
   write_dat(0x0F);
   write_dat(0x31);
   write_dat(0x2B);
   write_dat(0x0C);
   write_dat(0x0E);
   write_dat(0x08);
   write_dat(0x4E);
   write_dat(0xF1);
   write_dat(0x37);
   write_dat(0x07);
   write_dat(0x10);
   write_dat(0x03);
   write_dat(0x0E);
   write_dat(0x09);
   write_dat(0x00);

   write_cmd(0XE1);    //Set Gamma
   write_dat(0x00);
   write_dat(0x0E);
   write_dat(0x14);
   write_dat(0x03);
   write_dat(0x11);
   write_dat(0x07);
   write_dat(0x31);
   write_dat(0xC1);
   write_dat(0x48);
   write_dat(0x08);
```

```
    write_dat(0x0F);
    write_dat(0x0C);
    write_dat(0x31);
    write_dat(0x36);
    write_dat(0x0F);

    write_cmd(0x11);    //Exit Sleep
                    for( i = 0; i < 20000; i++) { i++;} //Wait a long time.

    write_cmd(0x29);    //Display on
    //writeCmd(0x2c);
}

//This function writes data to the LCD screen
void write_dat(unsigned char data)
{
        int FifoEmpty;
        int FifoBusy;
        //wait

        FifoEmpty = SSI0[0x0C/4] & 0x1;
        FifoBusy = (SSI0[0x0C/4] & 0x10) >> 4;
        while (FifoEmpty == 0x00 || FifoBusy == 0x01){                  //wait
                FifoEmpty = SSI0[0x0C/4] & 1;
                FifoBusy = (SSI0[0x0C/4] & 0x10) >> 4;
        }

        B0[0x3fc/4] = 0x1;              //set to data
        SSI0[0x08/4] = data;            //set data
}

//This function writes a command to the LCD screen
void write_cmd(unsigned char command)
{
        int FifoEmpty;
        int FifoBusy;
        //wait

        FifoEmpty = SSI0[0x0C/4] & 0x1;
        FifoBusy = (SSI0[0x0C/4] & 0x10) >> 4;
        while (FifoEmpty == 0x00 || FifoBusy == 0x01){                  //wait
                FifoEmpty = SSI0[0x0C/4] & 1;
                FifoBusy = (SSI0[0x0C/4] & 0x10) >> 4;
        }

        B0[0x3fc/4] = 0x0;              //set to data
        SSI0[0x08/4] = command;                 //set data

}
//this function parses 2 bytes and sends it
void write_dat_2(short data) {
        char first_byte = data >> 8;
        char second_byte = data;
        write_dat(first_byte);
```

```
	write_dat(second_byte);
}

//writes a single letter
void write_letter(short startX, short startY,
	unsigned int ASCII, short color, short backColor){

	unsigned int character;
	unsigned int value;
	int j;
	int i;
	int k;
	short mask;
	//define rows
	write_cmd(0x2b);
	write_dat_2(startY);
	write_dat_2(startY + 31);

	//define column
	write_cmd(0x2a);
	write_dat_2(startX);
	write_dat_2(startX + 23);

	character = 96 * (ASCII - 32);

	mask = 0x80;
	write_cmd(0x2C);

	for(i=0; i<=31; i++){			//height
		for(j=0; j<=2; j++){		//bytes in line
			value = Ubuntu[character++];
			for(k = 0; k <= 7; k++){
				//mean text is on
				if(value & mask){
					write_dat_2(color);
				}
				else{
					write_dat_2(backColor);
				}
				mask = mask >> 1;
			}
			//reset mask
			mask = 0x80;
		}
	}

}

//writes a word of length size
void write_word(short startX, short startY,
	char word[], int size, short color, short backColor){
		int i;
		int location = startX;
		for(i = 0; i < size; i++){
```

```
                    write_letter(location, startY, word[i], color, backColor);
                    location += 23;
            }
}

//draws a box to lcd
void draw_box(short startX, short startY, short width, short height, short color){
        int i;
        int size = width * height;
        //define rows
        write_cmd(0x2b);
        write_dat_2(startY);
        write_dat_2(startY + height - 1);

        //define column
        write_cmd(0x2a);
        write_dat_2(startX);
        write_dat_2(startX + width - 1);

        //set memory to write
        write_cmd(0x2c);

        //loop for size
        for(i = 0; i < size; i++){
                write_dat_2(color);
        }
}
```

Appendix D - Ubuntu.h

```
// Ubuntu.c
// Font type    : Full (95 characters)
// Font size    : 24x32 pixels
// Memory usage : 9124 bytes
// adapted from : http://www.henningkarlsen.com/electronics/r_fonts.php

unsigned char Ubuntu[] ={
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,  // <space>
0x00,0x00,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x0
0,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0
x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x18,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x1C,0x00,0x00,0x3E,0x00,0x00,0x3E,0x00,0x00,0x3E,0x00,0x00,0x1C,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,  // !
0x00,0x00,0x00,0x01,0xC0,0xE0,0x03,0xE1,0xF0,0x03,0xE1,0xF0,0x03,0xE1,0xF0,0x01,0xE0,0xF0,0x0
0,0xE0,0x70,0x01,0xC0,0xE0,0x03,0xC1,0xE0,0x0F,0x87,0xC0,0x0E,0x07,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,  // "
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x38,0x70,0x00,0x38,0x70,0x00,0x70,0xE0,0x00,0x70,0xE0,0x00,
0x70,0xE0,0x00,0x70,0xE0,0x0F,0xFF,0xF8,0x0F,0xFF,0xF8,0x0F,0xFF,0xF8,0x00,0xE1,0xC0,0x00,0x
E1,0xC0,0x00,0xE1,0xC0,0x01,0xC3,0x80,0x01,0xC3,0x80,0x01,0xC3,0x80,0x0F,0xFF,0xF8,0x0F,0xFF
,0xF8,0x0F,0xFF,0xF8,0x03,0x87,0x00,0x03,0x87,0x00,0x03,0x87,0x00,0x03,0x87,0x00,0x07,0x0E,0x0
0,0x07,0x0E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,  // #
0x00,0x00,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x7F,0x00,0x01,
0xFF,0xC0,0x03,0xFF,0xC0,0x07,0x80,0x80,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x80
,0x00,0x03,0xC0,0x00,0x03,0xF8,0x00,0x01,0xFE,0x00,0x00,0x7F,0x80,0x00,0x0F,0xC0,0x00,0x03,0xC
0,0x00,0x01,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x02,0x01,0xC0,0x07,0xFF,0xC0,0x
07,0xFF,0x80,0x01,0xFE,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x
00,0x00,  // $
0x00,0x00,0x00,0x07,0x80,0x70,0x0F,0xC0,0xE0,0x0C,0xC0,0xE0,0x18,0x61,0xC0,0x18,0x63,0x80,0x1
8,0x63,0x80,0x18,0x67,0x00,0x18,0x67,0x00,0x18,0x6E,0x00,0x0C,0xDC,0x00,0x0F,0xDC,0x00,0x07,0
xB8,0x00,0x00,0x3B,0xC0,0x00,0x77,0xE0,0x00,0x76,0x60,0x00,0xEC,0x30,0x01,0xCC,0x30,0x01,0xC
C,0x30,0x03,0x8C,0x30,0x03,0x8C,0x30,0x07,0x0C,0x30,0x0E,0x06,0x60,0x0E,0x07,0xE0,0x1C,0x03,0
xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,  // %
0x00,0x00,0x00,0x00,0x7C,0x00,0x00,0xFF,0x00,0x01,0xFF,0x00,0x03,0xC7,0x80,0x03,0x83,0x80,0x03
,0x83,0x80,0x03,0x83,0x80,0x03,0x87,0x00,0x01,0xCF,0x00,0x01,0xFE,0x00,0x00,0xFC,0x00,0x01,0xF
0,0x70,0x03,0xF8,0x70,0x03,0xBC,0x70,0x07,0x1E,0x60,0x0E,0x0F,0xE0,0x0E,0x07,0xE0,0x0E,0x03,0
xC0,0x0E,0x01,0xC0,0x0F,0x03,0xE0,0x07,0x8F,0xE0,0x07,0xFF,0x70,0x03,0xFE,0x70,0x00,0xF8,0x38
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,  // &
0x00,0x00,0x00,0x00,0x18,0x00,0x00,0x3C,0x00,0x00,0x1E,0x00,0x00,0x0E,0x00,0x00,0x07,0x00,0x00
,0x02,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,  // '
0x00,0x07,0x00,0x00,0x0F,0x80,0x00,0x0F,0x00,0x00,0x1E,0x00,0x00,0x3C,0x00,0x00,0x38,0x00,0x00
,0x70,0x00,0x00,0x70,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x01,0xC0,0x00,0x01,0xC
0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0
x00,0x01,0xC0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0x70,0x00,0x00,0x70,0x00,
0x00,0x38,0x00,0x00,0x3C,0x00,0x00,0x1E,0x00,0x00,0x0F,0x00,0x00,0x07,0x80,0x00,0x03,0x00,0x00,
0x01,0x00,  // (
0x00,0x60,0x00,0x00,0xF0,0x00,0x00,0x78,0x00,0x00,0x3C,0x00,0x00,0x1E,0x00,0x00,0x0E,0x00,0x00
,0x07,0x00,0x00,0x07,0x00,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x01,0xC0,0x00,0x01
,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0x
C0,0x00,0x01,0xC0,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x07,0x00,0x00,0x07,0x00,0x
00,0x0E,0x00,0x00,0x1E,0x00,0x00,0x3C,0x00,0x00,0x78,0x00,0x00,0xF0,0x00,0x00,0x60,0x00,0x00,0
x40,0x00,  // )
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x02,0x1C,0x20,0x03,0xC9,0xE0,0x07,0xF
F,0xF0,0x01,0xFF,0xC0,0x00,0x1C,0x00,0x00,0x36,0x00,0x00,0x77,0x00,0x00,0xF3,0x80,0x01,0xE3,0x
C0,0x00,0xE3,0x80,0x00,0x41,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,  // *
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x38,0x00,0x00,
0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x0F,0xFF,
0xE0,0x0F,0xFF,0xE0,0x0F,0xFF,0xE0,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x0
0,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,  // +
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0E,0x00,0x00,0x1F,0x00,0x00,0x1F,0x00,0x00,0x1F,0x00,0x00,
0x0F,0x00,0x00,0x07,0x00,0x00,0x0E,0x00,0x00,0x1C,0x00,0x00,0x78,0x00,0x00,0x70,0x00,0x00,0x00,
0x00,  // ,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,
0x00,0x00,0xFF,0x00,0x00,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,  // -
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1C,0x00,0x00,0x3E,0x00,0x00,0x3E,0x00,0x00
,0x3E,0x00,0x00,0x1C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,  // .
0x00,0x01,0xC0,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x07,0x00,0x00,0x07,0x00,0x00,
0x07,0x00,0x00,0x0E,0x00,0x00,0x0E,0x00,0x00,0x0E,0x00,0x00,0x0E,0x00,0x00,0x1C,0x00,0x00,0x1
C,0x00,0x00,0x1C,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x70,0x00,0x00,0x70,0x0
0,0x00,0x70,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x01,0xC0,0x00,0x
01,0xC0,0x00,0x01,0xC0,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x07,0x00,0x00,0x07,0x
00,0x00,  // /
```

```
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7C,0x00,0x01,0xFF,0x00,0x03,0xFF,0x80,0x03,0xC7,0x80,0x07
,0x83,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x30,0xE0,0x0E,0x
78,0xE0,0x0E,0x78,0xE0,0x0E,0x78,0xE0,0x0E,0x30,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,
0xE0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x83,0xC0,0x03,0xC7,0x80,0x03,0xFF,0x80,0x01,0xFF,0x0
0,0x00,0x7C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,  // 0
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x7C,0x00,0x00,0xFC,0x00,0x0
3,0xFC,0x00,0x07,0x9C,0x00,0x02,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0
x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C
,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x03,0xFF,0xC0,0x03,0xFF,0x
C0,0x03,0xFF,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,  // 1
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFC,0x00,0x03,0xFF,0x00,0x07,0xFF,0x80,0x07,0x87,0xC0,0x02
,0x03,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x03,0x80,0x00,0x0
7,0x80,0x00,0x07,0x00,0x00,0x0E,0x00,0x00,0x1C,0x00,0x00,0x38,0x00,0x00,0x70,0x00,0x00,0xE0,0x
00,0x01,0xC0,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x07,0x00,0x00,0x07,0xFF,0xE0,0x07,0xFF,0xE0,0
x07,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,  // 2
0x00,0x00,0x00,0x00,0x00,0x00,0x01,0xFC,0x00,0x0F,0xFF,0x00,0x0F,0xFF,0x80,0x06,0x07,0x80,0x00
,0x03,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x03,0xC0,0x00,0x07,0x80,0x00,0xF
F,0x00,0x00,0xFF,0x00,0x00,0xFF,0x80,0x00,0x07,0xC0,0x00,0x01,0xE0,0x00,0x00,0xE0,0x00,0x00,0x
E0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x01,0xE0,0x04,0x03,0xC0,0x0F,0xFF,0x80,0x0F,0xFF,0x00,
0x03,0xFC,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,  // 3
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x07,0x80,0x00,0x0F,0x80,0x00,0x1F,0x80,0x00,0x3F,0x80,0x00,
0x3B,0x80,0x00,0x73,0x80,0x00,0xF3,0x80,0x01,0xE3,0x80,0x01,0xC3,0x80,0x03,0x83,0x80,0x03,0x83
,0x80,0x07,0x03,0x80,0x0F,0x03,0x80,0x0E,0x03,0x80,0x1E,0x03,0x80,0x1F,0xFF,0xF0,0x1F,0xFF,0xF
0,0x1F,0xFF,0xF0,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x0
0,0x03,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,  // 4
0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xFF,0x80,0x03,0xFF,0x80,0x03,0xFF,0x80,0x03,0x00,0x00,0x03,
0x00,0x00,0x03,0x00,0x00,0x03,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0xF0,0x00,0x07,0xFE,
0x00,0x07,0xFF,0x00,0x00,0x1F,0x80,0x00,0x07,0x80,0x00,0x03,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC
0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x03,0xC0,0x0C,0x07,0x80,0x0F,0xFF,0x00,0x0F,0xFE,0x00,0
x03,0xF8,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,  // 5
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xC0,0x00,0x1F,0xC0,0x00,0x7F,0xC0,0x00,0xFC,0x00,0x0
1,0xE0,0x00,0x03,0xC0,0x00,0x03,0x80,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x06,0x7E,0x00,0x0F,0x
FF,0x80,0x0F,0xFF,0xC0,0x0F,0x03,0xC0,0x0E,0x01,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00
,0xE0,0x0E,0x00,0xE0,0x07,0x00,0xE0,0x07,0x01,0xC0,0x07,0xC3,0xC0,0x03,0xFF,0x80,0x01,0xFF,0x
00,0x00,0x7E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,  // 6
0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0xFF,0xE0,0x0F,0xFF,0xE0,0x0F,0xFF,0xE0,0x00,0x00,0xE0,0x0
0,0x01,0xC0,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x0E,0x00,0x00,0x0
E,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x70,0x
00,0x00,0x70,0x00,0x00,0x70,0x00,0x00,0x70,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x
00,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,  // 7
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x7C,0x00,0x01,0xFF,0x00,0x03,0xFF,0x80,0x03,0xC7,0xC0,0x0
7,0x03,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x81,0xC0,0x03,0xC3,0x80,0x03,0
xF7,0x00,0x01,0xFE,0x00,0x01,0xFF,0x80,0x03,0xCF,0xC0,0x07,0x03,0xC0,0x06,0x01,0xE0,0x0E,0x00
,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0F,0x01,0xE0,0x07,0x83,0xC0,0x07,0xFF,0xC0,0x03,0xFF,0x
80,0x00,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,  // 8
```

```
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFC,0x00,0x01,0xFF,0x00,0x03,0xFF,0x80,0x07,0x87,0xC0,0x07
,0x01,0xC0,0x0E,0x01,0xC0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0F,0x
00,0xE0,0x07,0x81,0xE0,0x07,0xFF,0xE0,0x03,0xFF,0xE0,0x00,0xFE,0xE0,0x00,0x01,0xC0,0x00,0x01,
0xC0,0x00,0x03,0xC0,0x00,0x07,0x80,0x00,0x0F,0x00,0x00,0x7E,0x00,0x07,0xFC,0x00,0x07,0xF8,0x0
0,0x07,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,  // 9
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x38,0x00,0x00,0x7C,0x00,0x00,0x7C,0x00,0x00,0x7C,0x00,0x00,0x38
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x38,0x00,0x00,0x7C,0x00,0x00,0x7C,0x00,0x00,0x7C,0x00,0x00,0x38,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,  // :
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x38,0x00,0x00,0x7C,0x00,0x00,0x7C,0x00,0x00,0x7C,0x00,0x00,0x38
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x38,0x00,0x00,0x7C,0x00,0x00,0x7C,0x00,0x00,0x7C,0x00,0x00,0x3C,0x00,0x0
0,0x1C,0x00,0x00,0x38,0x00,0x00,0x70,0x00,0x01,0xE0,0x00,0x01,0xC0,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,  // ;
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x40,0x00,0x01,0xE0,0x00,0x0F,0xE0,0x00,0x3F,0xC0,0x00,0xF
E,0x00,0x03,0xF8,0x00,0x0F,0xC0,0x00,0x0F,0x00,0x00,0x0F,0xC0,0x00,0x07,0xF8,0x00,0x00,0xFE,0x
00,0x00,0x3F,0xC0,0x00,0x0F,0xE0,0x00,0x01,0xE0,0x00,0x00,0x40,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,  // <
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0xFF,0xE0,0x0F,0xFF,0xE0,0x0F,0xF
F,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0xFF,0x
E0,0x0F,0xFF,0xE0,0x0F,0xFF,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,  // =
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x04,0x00,0x00,0x0F,0x00,0x00,0x0F,0xE0,0x00,0x07,0xF8,0x00,0x00,0xFE
,0x00,0x00,0x3F,0x80,0x00,0x07,0xE0,0x00,0x01,0xE0,0x00,0x07,0xE0,0x00,0x3F,0xC0,0x00,0xFE,0x0
0,0x07,0xF8,0x00,0x0F,0xE0,0x00,0x0F,0x00,0x00,0x04,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,  // >
0x00,0x00,0x00,0x00,0xFE,0x00,0x03,0xFF,0x80,0x03,0xFF,0xC0,0x01,0x03,0xE0,0x00,0x00,0xE0,0x0
0,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x01,0xC0,0x00,0x03,0x80,0x00,0x07,0x00,0x00,0x
0E,0x00,0x00,0x1C,0x00,0x00,0x3C,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x38,0x00,0x00,0x7C,0x00,0x00,0x7C,0x00,0x00,0x7C,0x00,0x00,0x38,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,  // ?

0x00,0x00,0x00,0x00,0x3F,0x00,0x00,0xFF,0x80,0x01,0xFF,0xC0,0x01,0xE1,0xE0,0x03,0x80,0xE0,0x0
3,0x80,0x70,0x07,0x00,0x70,0x07,0x00,0x70,0x07,0x07,0xF0,0x0E,0x1F,0xF0,0x0E,0x3F,0xF0,0x0E,0x
3C,0x70,0x0E,0x78,0x70,0x0E,0x70,0x70,0x0E,0x70,0x70,0x0E,0x70,0x70,0x0E,0x70,0x70,0x0E,0x78,0
x70,0x0E,0x38,0x70,0x07,0x3F,0xF0,0x07,0x1F,0xF0,0x07,0x07,0xE0,0x07,0x80,0x00,0x03,0xC0,0x00,
0x03,0xC0,0x00,0x01,0xF0,0x00,0x00,0xFF,0xC0,0x00,0x7F,0xC0,0x00,0x1F,0xC0,0x00,0x00,0x00,0x0
0,0x00,0x00,  // @
0x00,0x00,0x00,0x00,0x7E,0x00,0x00,0x7E,0x00,0x00,0x7E,0x00,0x00,0xFF,0x00,0x00,0xF7,0x00,0x00
,0xE7,0x00,0x00,0xE7,0x00,0x01,0xE7,0x80,0x01,0xC3,0x80,0x01,0xC3,0x80,0x03,0xC3,0xC0,0x03,0x
C1,0xC0,0x03,0x81,0xC0,0x03,0x81,0xC0,0x07,0x81,0xE0,0x07,0xFF,0xE0,0x07,0xFF,0xE0,0x07,0xFF,
0xE0,0x0F,0x00,0xF0,0x0E,0x00,0x70,0x0E,0x00,0x70,0x0E,0x00,0x70,0x1E,0x00,0x78,0x1C,0x00,0x3
```

```
8,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,  // A
0x00,0x00,0x00,0x00,0x00,0x00,0x0F,0xFC,0x00,0x0F,0xFF,0x00,0x0F,0xFF,0x80,0x0E,0x07,0xC0,0x0
E,0x03,0xC0,0x0E,0x01,0xC0,0x0E,0x01,0xC0,0x0E,0x01,0xC0,0x0E,0x03,0x80,0x0E,0x07,0x80,0x0F,0
xFF,0x00,0x0F,0xFF,0x00,0x0F,0xFF,0x80,0x0E,0x03,0xC0,0x0E,0x01,0xE0,0x0E,0x00,0xE0,0x0E,0x0
0,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x01,0xE0,0x0E,0x07,0xC0,0x0F,0xFF,0xC0,0x0F,0xFF,
0x00,0x0F,0xFC,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,  // B
0x00,0x00,0x00,0x00,0x3F,0x80,0x00,0x7F,0xE0,0x01,0xFF,0xE0,0x03,0xE0,0x40,0x03,0xC0,0x00,0x0
7,0x80,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x0E,0x00,0x00,0x0E,0x00,0x00,0x0E,0x00,0x00,0x0E,0x
00,0x00,0x0E,0x00,0x00,0x0E,0x00,0x00,0x0E,0x00,0x00,0x0E,0x00,0x00,0x0F,0x00,0x00,0x07,0x00,0
x00,0x07,0x80,0x00,0x03,0xC0,0x00,0x03,0xE0,0x40,0x01,0xFF,0xE0,0x00,0xFF,0xE0,0x00,0x3F,0x80,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,  // C
0x00,0x00,0x00,0x0F,0xF0,0x00,0x0F,0xFE,0x00,0x0F,0xFF,0x00,0x0E,0x0F,0x80,0x0E,0x03,0x80,0x0
E,0x03,0xC0,0x0E,0x01,0xC0,0x0E,0x01,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,
0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x01,0xE0,0x0E,0x
01,0xC0,0x0E,0x03,0xC0,0x0E,0x03,0x80,0x0E,0x0F,0x80,0x0F,0xFF,0x00,0x0F,0xFE,0x00,0x0F,0xF0,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,  // D
0x00,0x00,0x00,0x07,0xFF,0xC0,0x07,0xFF,0xC0,0x07,0xFF,0xC0,0x07,0x00,0x00,0x07,0x00,0x00,0x0
7,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0xFF,0x80,0x07,0xF
F,0x80,0x07,0xFF,0x80,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x0
0,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0xFF,0xE0,0x07,0xFF,0xE0,0x07,0xFF,0xE0,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,  // E
0x00,0x00,0x00,0x07,0xFF,0xE0,0x07,0xFF,0xE0,0x07,0xFF,0xE0,0x07,0x00,0x00,0x07,0x00,0x00,0x0
7,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0xFF,0xC0,0x07,0xF
F,0xC0,0x07,0xFF,0xC0,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x
00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,  // F
0x00,0x00,0x00,0x00,0x3F,0x80,0x00,0xFF,0xC0,0x01,0xFF,0xE0,0x03,0xE0,0xC0,0x03,0xC0,0x40,0x0
7,0x80,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x0E,0x00,0x00,0x0E,0x00,0x00,0x0E,0x00,0x00,0x0E,0x
00,0x00,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0F,0x00,0xE0,0x07,0x00,
0xE0,0x07,0x80,0xE0,0x03,0xC0,0xE0,0x03,0xE0,0xE0,0x01,0xFF,0xE0,0x00,0xFF,0xE0,0x00,0x3F,0x
80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,  // G
0x00,0x00,0x00,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0
E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0F,0xFF,0xE0,0x0F,
0xFF,0xE0,0x0F,0xFF,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x
00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,
0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,  // H
0x00,0x00,0x00,0x07,0xFF,0xC0,0x07,0xFF,0xC0,0x07,0xFF,0xC0,0x00,0x38,0x00,0x00,0x38,0x00,0x0
0,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x3
8,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x0
0,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x07,0xFF,0xC0,0x07,0xFF,0xC0,0x07,0xFF,0xC0,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,  // I
0x00,0x00,0x00,0x01,0xFF,0xC0,0x01,0xFF,0xC0,0x01,0xFF,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x0
0,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0
x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01
,0xC0,0x00,0x01,0xC0,0x04,0x03,0xC0,0x07,0x07,0x80,0x0F,0xFF,0x00,0x07,0xFE,0x00,0x01,0xFC,0x
```

```
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,  // J
0x00,0x00,0x00,0x0E,0x01,0xE0,0x0E,0x03,0xC0,0x0E,0x03,0x80,0x0E,0x07,0x80,0x0E,0x0F,0x00,0x0
E,0x0E,0x00,0x0E,0x1C,0x00,0x0E,0x3C,0x00,0x0E,0x78,0x00,0x0E,0xF0,0x00,0x0F,0xE0,0x00,0x0F,0
xE0,0x00,0x0F,0xF0,0x00,0x0E,0xF8,0x00,0x0E,0x7C,0x00,0x0E,0x3E,0x00,0x0E,0x1E,0x00,0x0E,0x0
F,0x00,0x0E,0x07,0x80,0x0E,0x03,0xC0,0x0E,0x03,0xC0,0x0E,0x01,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0
xF0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,  // K
0x00,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,
0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,
0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,
0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0xFF,0xE0,0x07,0xFF,0xE0,0x07,0xFF,0xE0,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,  // L
0x00,0x00,0x00,0x03,0x00,0x60,0x03,0x80,0xF0,0x07,0x80,0xF0,0x07,0x80,0xF0,0x06,0xC1,0xB0,0x06
,0xC1,0xB0,0x06,0xE3,0xB0,0x06,0x63,0x30,0x06,0x63,0x30,0x06,0x77,0x30,0x06,0x36,0x30,0x06,0x3
6,0x30,0x06,0x3E,0x30,0x06,0x1C,0x38,0x0E,0x1C,0x38,0x0E,0x00,0x38,0x0E,0x00,0x38,0x0E,0x00,0x
38,0x0E,0x00,0x38,0x0E,0x00,0x38,0x0E,0x00,0x38,0x0E,0x00,0x38,0x0E,0x00,0x38,0x0E,0x00,0x38,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,  // M
0x00,0x00,0x00,0x07,0x80,0x70,0x07,0xC0,0x70,0x07,0xC0,0x70,0x07,0xE0,0x70,0x07,0xE0,0x70,0x07
,0xF0,0x70,0x07,0x70,0x70,0x07,0x70,0x70,0x07,0x38,0x70,0x07,0x38,0x70,0x07,0x1C,0x70,0x07,0x1
C,0x70,0x07,0x0E,0x70,0x07,0x0E,0x70,0x07,0x06,0x70,0x07,0x07,0x70,0x07,0x07,0x70,0x07,0x03,0x
F0,0x07,0x03,0xF0,0x07,0x01,0xF0,0x07,0x01,0xF0,0x07,0x00,0xF0,0x07,0x00,0xF0,0x07,0x00,0xF0,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,  // N
0x00,0x00,0x00,0x00,0x7C,0x00,0x01,0xFF,0x00,0x03,0xFF,0x80,0x07,0x83,0xC0,0x0F,0x01,0xE0,0x0
E,0x00,0xE0,0x0E,0x00,0xE0,0x1C,0x00,0xF0,0x1C,0x00,0x70,0x1C,0x00,0x70,0x1C,0x00,0x70,0x1C,0
x00,0x70,0x1C,0x00,0x70,0x1C,0x00,0x70,0x1C,0x00,0x70,0x1C,0x00,0x70,0x1C,0x00,0xF0,0x0E,0x00
,0xE0,0x0E,0x00,0xE0,0x0F,0x01,0xE0,0x07,0x83,0xC0,0x03,0xFF,0x80,0x01,0xFF,0x00,0x00,0x7C,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,  // O

0x00,0x00,0x00,0x07,0xFC,0x00,0x07,0xFF,0x00,0x07,0xFF,0x80,0x07,0x03,0xC0,0x07,0x01,0xE0,0x0
7,0x00,0xE0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x01,0xE0,0x07,0x
03,0xC0,0x07,0xFF,0x80,0x07,0xFF,0x00,0x07,0xFC,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0
x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,  // P
0x00,0x00,0x00,0x00,0x7C,0x00,0x01,0xFF,0x00,0x03,0xFF,0x80,0x07,0x83,0xC0,0x0F,0x01,0xE0,0x0
E,0x00,0xE0,0x0E,0x00,0xE0,0x1C,0x00,0xF0,0x1C,0x00,0x70,0x1C,0x00,0x70,0x1C,0x00,0x70,0x1C,0
x00,0x70,0x1C,0x00,0x70,0x1C,0x00,0x70,0x1C,0x00,0x70,0x1C,0x00,0x70,0x1C,0x00,0xF0,0x0E,0x00
,0xE0,0x0E,0x00,0xE0,0x0F,0x01,0xE0,0x07,0x83,0xC0,0x03,0xFF,0x80,0x01,0xFF,0x00,0x00,0xFE,0x
00,0x00,0x38,0x00,0x00,0x3C,0x00,0x00,0x1F,0x00,0x00,0x0F,0xE0,0x00,0x07,0xE0,0x00,0x00,0xC0,0
x00,0x00,0x00,  // Q
0x00,0x00,0x00,0x0F,0xF8,0x00,0x0F,0xFE,0x00,0x0F,0xFF,0x00,0x0E,0x07,0x80,0x0E,0x03,0xC0,0x0
E,0x01,0xC0,0x0E,0x01,0xC0,0x0E,0x01,0xC0,0x0E,0x01,0xC0,0x0E,0x01,0xC0,0x0E,0x03,0x80,0x0E,
0x07,0x80,0x0F,0xFF,0x00,0x0F,0xFE,0x00,0x0F,0xFC,0x00,0x0E,0x1E,0x00,0x0E,0x0E,0x00,0x0E,0x
07,0x00,0x0E,0x07,0x00,0x0E,0x03,0x80,0x0E,0x03,0x80,0x0E,0x01,0xC0,0x0E,0x01,0xC0,0x0E,0x00,
0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,  // R
0x00,0x00,0x00,0x00,0x7F,0x00,0x01,0xFF,0xC0,0x03,0xFF,0xC0,0x03,0xC0,0x80,0x07,0x00,0x00,0x0
7,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x80,0x00,0x03,0xE0,0x00,0x01,0xF8,0x00,0x00,0xF
E,0x00,0x00,0x3F,0x80,0x00,0x0F,0xC0,0x00,0x03,0xC0,0x00,0x01,0xE0,0x00,0x00,0xE0,0x00,0x00,0x
```

```
E0,0x00,0x00,0xE0,0x04,0x01,0xE0,0x0F,0x03,0xC0,0x0F,0xFF,0xC0,0x07,0xFF,0x80,0x01,0xFE,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,  // S
0x00,0x00,0x00,0x0F,0xFF,0xE0,0x0F,0xFF,0xE0,0x0F,0xFF,0xE0,0x00,0x38,0x00,0x00,0x38,0x00,0x0
0,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x3
8,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x0
0,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,  // T
0x00,0x00,0x00,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0
E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,
0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x
00,0xE0,0x0F,0x01,0xE0,0x07,0x01,0xC0,0x07,0x83,0xC0,0x03,0xFF,0x80,0x01,0xFF,0x00,0x00,0xFE,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,  // U
0x00,0x00,0x00,0x1C,0x00,0x70,0x1C,0x00,0x70,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0
E,0x00,0xE0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x03,0x83,0x80,0x03,0x
83,0x80,0x03,0x83,0x80,0x03,0x87,0x80,0x01,0xC7,0x00,0x01,0xC7,0x00,0x01,0xC7,0x00,0x00,0xEE,0
x00,0x00,0xEE,0x00,0x00,0xEE,0x00,0x00,0x7C,0x00,0x00,0x7C,0x00,0x00,0x7C,0x00,0x00,0x78,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,  // V
0x00,0x00,0x00,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0
E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x38,0xE0,0x0E,0x38,0xE0,0x0E,
0x38,0xE0,0x0E,0x6C,0xE0,0x0E,0x6C,0xE0,0x0E,0x6C,0xE0,0x0E,0xC6,0xE0,0x0E,0xC6,0xE0,0x0E,0
xC6,0xE0,0x0F,0x83,0xE0,0x0F,0x83,0xE0,0x0F,0x83,0xE0,0x0F,0x01,0xE0,0x07,0x01,0xC0,0x07,0x01
,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,  // W
0x00,0x00,0x00,0x1E,0x00,0xF0,0x0E,0x00,0xE0,0x0F,0x01,0xE0,0x07,0x01,0xC0,0x03,0x83,0x80,0x0
3,0x83,0x80,0x01,0xC7,0x00,0x01,0xEF,0x00,0x00,0xEE,0x00,0x00,0xFE,0x00,0x00,0x7C,0x00,0x00,0x
38,0x00,0x00,0x38,0x00,0x00,0x7C,0x00,0x00,0xFE,0x00,0x00,0xEE,0x00,0x01,0xC7,0x00,0x03,0xC7,
0x80,0x03,0x83,0x80,0x07,0x83,0xC0,0x07,0x01,0xC0,0x0F,0x01,0xE0,0x0E,0x00,0xE0,0x1E,0x00,0xF
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,  // X
0x00,0x00,0x00,0x1C,0x00,0x70,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0F,0x01,0xE0,0x07,0x01,0xC0,0x0
7,0x83,0xC0,0x03,0x83,0x80,0x03,0xC3,0x80,0x01,0xC7,0x00,0x01,0xEF,0x00,0x00,0xEE,0x00,0x00,0x
FE,0x00,0x00,0x7C,0x00,0x00,0x7C,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0
x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,  // Y
0x00,0x00,0x00,0x07,0xFF,0xE0,0x07,0xFF,0xE0,0x07,0xFF,0xE0,0x00,0x01,0xC0,0x00,0x03,0xC0,0x0
0,0x03,0x80,0x00,0x07,0x00,0x00,0x0F,0x00,0x00,0x0E,0x00,0x00,0x1E,0x00,0x00,0x3C,0x00,0x00,0x
38,0x00,0x00,0x78,0x00,0x00,0x70,0x00,0x00,0xE0,0x00,0x01,0xE0,0x00,0x01,0xC0,0x00,0x03,0xC0,0
x00,0x03,0x80,0x00,0x07,0x80,0x00,0x07,0x00,0x00,0x0F,0xFF,0xE0,0x0F,0xFF,0xE0,0x0F,0xFF,0xE0,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,  // Z
0x01,0xFF,0x00,0x01,0xFF,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x0
1,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0
xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0
,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x
00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xFF,0x00,
0x01,0xFF,0x00,  // [
0x03,0x80,0x00,0x03,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x0
0,0xE0,0x00,0x00,0x70,0x00,0x00,0x70,0x00,0x00,0x70,0x00,0x00,0x70,0x00,0x00,0x38,0x00,0x00,0x3
8,0x00,0x00,0x38,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x0E,0x00,0x00,0x0E,0x
00,0x00,0x0E,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x03,0x80,0x
00,0x03,0x80,0x00,0x03,0x80,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x00,0x00,0xE0,0x00,0
x00,0xE0,  // <backslash>
0x00,0xFF,0x80,0x00,0xFF,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,
0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,
0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,
0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,
0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0xFF,0x80,0x00,0xFF,
0x80,  // ]
0x00,0x00,0x00,0x00,0x38,0x00,0x00,0x7C,0x00,0x00,0x7C,0x00,0x00,0xEE,0x00,0x00,0xEE,0x00,0x0
1,0xC7,0x00,0x01,0xC7,0x00,0x03,0xC7,0x80,0x03,0x83,0x80,0x07,0x83,0xC0,0x07,0x01,0xC0,0x0F,0x
01,0xE0,0x02,0x00,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,  // ^
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x3F,0xFF,0xF8,0x3F,0xFF,0xF8,0x3F
,0xFF,0xF8,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,  // _

0x00,0x00,0x00,0x00,0x7C,0x00,0x00,0xFE,0x00,0x01,0xC7,0x00,0x01,0x83,0x00,0x01,0x83,0x00,0x01
,0x83,0x00,0x01,0xC7,0x00,0x00,0xFE,0x00,0x00,0x7C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,  // `
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0xFE,0x00,0x01,0xFF,0x80,0x01,0xFF,0xC0,0x01,0x83,0xE0,0x00,0x0
1,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x7F,0xE0,0x01,0xFF,0xE0,0x03,0xFF,0xE0,0x07,0xC0,0
xE0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x80,0xE0,0x03,0xFF,0xE0,0x01,0xFF,0xE0
,0x00,0x7F,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,  // a
0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,
0x00,0x00,0x07,0x00,0x00,0x07,0x7E,0x00,0x07,0xFF,0x00,0x07,0xFF,0x80,0x07,0x83,0xC0,0x07,0x01
,0xC0,0x07,0x01,0xE0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x00,0xE
0,0x07,0x00,0xE0,0x07,0x01,0xE0,0x07,0x01,0xC0,0x07,0x07,0xC0,0x07,0xFF,0x80,0x07,0xFF,0x00,0x
03,0xFC,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,  // b
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x3F,0xC0,0x00,0xFF,0xE0,0x03,0xFF,0xE0,0x03,0xE0,0x40,0x07,0x8
0,0x00,0x07,0x00,0x00,0x0E,0x00,0x00,0x0E,0x00,0x00,0x0E,0x00,0x00,0x0E,0x00,0x00,0x0E,0x00,0x
00,0x0E,0x00,0x00,0x07,0x00,0x00,0x07,0x80,0x00,0x03,0xE0,0x60,0x03,0xFF,0xE0,0x00,0xFF,0xE0,0
x00,0x3F,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,  // c
0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x0
0,0x00,0xE0,0x00,0x00,0xE0,0x00,0x7E,0xE0,0x00,0xFF,0xE0,0x01,0xFF,0xE0,0x03,0xC1,0xE0,0x03,0
x80,0xE0,0x07,0x80,0xE0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x00,
0xE0,0x07,0x00,0xE0,0x07,0x80,0xE0,0x03,0x80,0xE0,0x03,0xE0,0xE0,0x01,0xFF,0xE0,0x00,0xFF,0xE
0,0x00,0x3F,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,  // d
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x7E,0x00,0x01,0xFF,0x00,0x03,0xFF,0x80,0x07,0xC3,0xC0,0x07,0x0
```

```
1,0xC0,0x07,0x00,0xE0,0x0E,0x00,0xE0,0x0F,0xFF,0xE0,0x0F,0xFF,0xE0,0x0F,0xFF,0xE0,0x0E,0x00,
0x00,0x0E,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0xC0,0xC0,0x03,0xFF,0xC0,0x01,0xFF,0xC
0,0x00,0x7F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,  // e
0x00,0x0F,0xC0,0x00,0x3F,0xF0,0x00,0x7F,0xF0,0x00,0x78,0x30,0x00,0xF0,0x00,0x00,0xE0,0x00,0x00
,0xE0,0x00,0x00,0xE0,0x00,0x0F,0xFF,0xC0,0x0F,0xFF,0xC0,0x0F,0xFF,0xC0,0x00,0xE0,0x00,0x00,0x
E0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0
x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,
0x00,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,  // f
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x7F,0x00,0x01,0xFF,0xC0,0x03,0xFF,0xC0,0x07,0x81,0xC0,0x07,0x01,0xC0,0x0F,0x0
1,0xC0,0x0E,0x01,0xC0,0x0E,0x01,0xC0,0x0E,0x01,0xC0,0x0E,0x01,0xC0,0x0E,0x01,0xC0,0x0E,0x01,
0xC0,0x07,0x01,0xC0,0x07,0x83,0xC0,0x03,0xFF,0xC0,0x01,0xFF,0xC0,0x00,0xFD,0xC0,0x00,0x01,0x
C0,0x00,0x01,0xC0,0x00,0x03,0xC0,0x06,0x07,0x80,0x07,0xFF,0x80,0x07,0xFF,0x00,0x03,0xFC,0x00,
0x00,0x00,0x00,  // g
0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,
0x00,0x00,0x07,0x00,0x00,0x07,0xFC,0x00,0x07,0xFF,0x00,0x07,0xFF,0x80,0x07,0x07,0x80,0x07,0x03
,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x01,0x
C0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,
0x07,0x01,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,  // h
0x00,0x00,0x00,0x00,0x60,0x00,0x00,0xF0,0x00,0x00,0xF0,0x00,0x00,0x60,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x0F,0xF0,0x00,0x0F,0xF0,0x00,0x0F,0xF0,0x00,0x00,0x70,0x00,0x00,0x70,
0x00,0x00,0x70,0x00,0x00,0x70,0x00,0x00,0x70,0x00,0x00,0x70,0x00,0x00,0x70,0x00,0x00,0x70,0x00,
0x00,0x70,0x00,0x00,0x70,0x00,0x00,0x70,0x00,0x00,0x78,0x20,0x00,0x3F,0xE0,0x00,0x3F,0xE0,0x00,
0x0F,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,  // i
0x00,0x00,0x00,0x00,0x06,0x00,0x00,0x0F,0x00,0x00,0x0F,0x00,0x00,0x06,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x03,0xFF,0x80,0x03,0xFF,0x80,0x03,0xFF,0x80,0x00,0x03,0x80,0x00,0x03,
0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,
0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,
0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x06,0x0F,0x00,0x0F,0xFF,0x00,0x0F,0xFE,0x00,0x03,0xF8
,0x00,  // j
0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,
0x00,0x00,0x07,0x00,0x00,0x07,0x01,0xE0,0x07,0x03,0xC0,0x07,0x07,0x80,0x07,0x0F,0x00,0x07,0x1E
,0x00,0x07,0x3C,0x00,0x07,0x78,0x00,0x07,0xF0,0x00,0x07,0xE0,0x00,0x07,0xF0,0x00,0x07,0x78,0x0
0,0x07,0x1C,0x00,0x07,0x0F,0x00,0x07,0x07,0x00,0x07,0x03,0x80,0x07,0x03,0xC0,0x07,0x01,0xE0,0x
07,0x00,0xF0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,  // k
0x07,0xF8,0x00,0x07,0xF8,0x00,0x07,0xF8,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,
0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,
0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,
0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x38,0x00,0x00,0x3C,0x10,0x00,0x1F,0xF0,0x00,0x1F,0xF0,0x00
,0x07,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,  // l
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x07,0xE3,0xC0,0x0F,0xFF,0xE0,0x0F,0xFF,0xE0,0x0E,0x3C,0xF0,0x0E,0x
1C,0x70,0x0E,0x1C,0x70,0x0E,0x1C,0x70,0x0E,0x1C,0x70,0x0E,0x1C,0x70,0x0E,0x1C,0x70,0x0E,0x1
C,0x70,0x0E,0x00,0x70,0x0E,0x00,0x70,0x0E,0x00,0x70,0x0E,0x00,0x70,0x0E,0x00,0x70,0x0E,0x00,0x
70,0x0E,0x00,0x70,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,  // m
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0xFE,0x00,0x03,0xFF,0x80,0x03,0xFF,0xC0,0x03,0x83,0xC0,0x03,0x8
```

```
1,0xE0,0x03,0x80,0xE0,0x03,0x80,0xE0,0x03,0x80,0xE0,0x03,0x80,0xE0,0x03,0x80,0xE0,0x03,0x80,0x
E0,0x03,0x80,0xE0,0x03,0x80,0xE0,0x03,0x80,0xE0,0x03,0x80,0xE0,0x03,0x80,0xE0,0x03,0x80,0xE0,0
x03,0x80,0xE0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,  // n
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x7C,0x00,0x01,0xFF,0x00,0x03,0xFF,0x80,0x07,0x83,0xC0,0x07,0x01
,0xC0,0x0F,0x01,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0E,0x00,0x
E0,0x0E,0x00,0xE0,0x07,0x01,0xE0,0x07,0x01,0xC0,0x07,0x83,0xC0,0x03,0xFF,0x80,0x01,0xFF,0x00,
0x00,0x7C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,  // o

0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x01,0xFC,0x00,0x07,0xFF,0x00,0x07,0xFF,0x80,0x07,0x07,0xC0,0x07,0x01
,0xC0,0x07,0x01,0xC0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x00,0xE
0,0x07,0x00,0xE0,0x07,0x01,0xE0,0x07,0x01,0xC0,0x07,0x83,0xC0,0x07,0xFF,0x80,0x07,0xFF,0x00,0x
07,0x7C,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x
00,0x00,  // p
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x3F,0x80,0x00,0xFF,0xE0,0x01,0xFF,0xE0,0x03,0xE0,0xE0,0x03,0x8
0,0xE0,0x03,0x80,0xE0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x00,0x
E0,0x07,0x00,0xE0,0x07,0x80,0xE0,0x03,0x80,0xE0,0x03,0xC1,0xE0,0x01,0xFF,0xE0,0x00,0xFF,0xE0,
0x00,0x3E,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x0
0,0x00,0xE0,  // q
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0xC0,0x07,0xFF,0xC0,0x07,0xFF,0xC0,0x07,0x80,0x80,0x07,0x0
0,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x0
0,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x07,0x00,0x00,0x0
7,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,  // r
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0x00,0x01,0xFF,0x80,0x03,0xFF,0x80,0x07,0x81,0x80,0x07,0x00,
0x00,0x07,0x00,0x00,0x07,0xC0,0x00,0x03,0xF0,0x00,0x01,0xFE,0x00,0x00,0x7F,0x00,0x00,0x1F,0x80
,0x00,0x03,0xC0,0x00,0x01,0xC0,0x00,0x01,0xC0,0x06,0x03,0xC0,0x07,0xFF,0x80,0x07,0xFF,0x80,0x0
1,0xFE,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,  // s
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,
0xE0,0x00,0x00,0xE0,0x00,0x0F,0xFF,0xE0,0x0F,0xFF,0xE0,0x0F,0xFF,0xE0,0x00,0xE0,0x00,0x00,0x
E0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0
x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xF0,0x20,0x00,0x7F,0xE0,0x00,0x7F,0xE0,
0x00,0x1F,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,  // t
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x0
1,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x01,0
xC0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x81,0xC0,0x03,0xC1,0xC0,0x03,0xFF,0xC0,0x01,0xFF,0xC
0,0x00,0x7F,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,  // u
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x1C,0x00,0x70,0x1E,0x00,0xF0,0x0E,0x00,0xE0,0x0E,0x00,0xE0,0x0F,0x0
1,0xE0,0x07,0x01,0xC0,0x07,0x01,0xC0,0x07,0x83,0xC0,0x03,0x83,0x80,0x03,0x83,0x80,0x03,0xC7,0x
00,0x01,0xC7,0x00,0x01,0xEF,0x00,0x00,0xEE,0x00,0x00,0xFE,0x00,0x00,0x7C,0x00,0x00,0x7C,0x00,
0x00,0x7C,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,  // v
```

```
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x1C,0x00,0x70,0x1C,0x00,0x70,0x1C,0x00,0x70,0x1C,0x00,0x70,0x1C,0x1
0,0x70,0x0C,0x38,0xE0,0x0E,0x38,0xE0,0x0E,0x38,0xE0,0x0E,0x28,0xE0,0x0E,0x6C,0xE0,0x0E,0x6C,
0xE0,0x06,0x44,0xC0,0x06,0xC6,0xC0,0x06,0xC7,0xC0,0x07,0x87,0xC0,0x07,0x83,0xC0,0x07,0x83,0x
80,0x03,0x03,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,  // w
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x1E,0x01,0xE0,0x0F,0x03,0xC0,0x07,0x03,0x80,0x03,0x87,0x80,0x03,0xC
F,0x00,0x01,0xEE,0x00,0x00,0xFC,0x00,0x00,0x7C,0x00,0x00,0x78,0x00,0x00,0x7C,0x00,0x00,0xFE,0
x00,0x01,0xEE,0x00,0x01,0xC7,0x00,0x03,0xC3,0x80,0x07,0x83,0xC0,0x0F,0x01,0xC0,0x0E,0x01,0xE0
,0x1E,0x00,0xF0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,  // x
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x0E,0x00,0x70,0x0F,0x00,0xF0,0x07,0x00,0xE0,0x07,0x00,0xE0,0x07,0x80
,0xE0,0x03,0x81,0xE0,0x03,0x81,0xC0,0x03,0xC1,0xC0,0x01,0xC3,0xC0,0x01,0xC3,0x80,0x01,0xE3,0x
80,0x00,0xE7,0x80,0x00,0xF7,0x80,0x00,0x77,0x00,0x00,0x7F,0x00,0x00,0x3E,0x00,0x00,0x3E,0x00,0
x00,0x1E,0x00,0x00,0x3C,0x00,0x00,0x3C,0x00,0x00,0x78,0x00,0x0F,0xF0,0x00,0x0F,0xF0,0x00,0x0F,
0xC0,0x00,  // y
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x07,0xFF,0xC0,0x07,0xFF,0xC0,0x07,0xFF,0xC0,0x00,0x03,0x80,0x00,0x0
7,0x80,0x00,0x0F,0x00,0x00,0x1E,0x00,0x00,0x1C,0x00,0x00,0x38,0x00,0x00,0x78,0x00,0x00,0xF0,0x
00,0x00,0xE0,0x00,0x01,0xE0,0x00,0x03,0xC0,0x00,0x03,0x80,0x00,0x07,0xFF,0xC0,0x07,0xFF,0xC0,
0x07,0xFF,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,  // z
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x70,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,
0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE
0,0x00,0x00,0xE0,0x00,0x01,0xC0,0x00,0x0F,0x80,0x00,0x0F,0x80,0x00,0x01,0xC0,0x00,0x00,0xE0,0x
00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0
x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0xE0,0x00,0x00,0x70,0x00,0x00,0x7F,0x80,0x00,
0x1F,0x80,  // {
0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x0
0,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0
x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C
,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x
00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,0x00,0x1C,0x00,
0x00,0x1C,0x00,  // |
0x00,0xFC,0x00,0x00,0xFF,0x00,0x00,0x07,0x00,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,
0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,
0x80,0x00,0x03,0x80,0x00,0x01,0xC0,0x00,0x00,0xF8,0x00,0x00,0xF8,0x00,0x01,0xC0,0x00,0x03,0x80
,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,
0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x03,0x80,0x00,0x07,0x00,0x00,0xFF,0x00,0x00,0xFC,
0x00,  // }
0x00,0x00,0x00,0x01,0xE0,0x20,0x03,0xF0,0x30,0x07,0xFC,0x70,0x0E,0x3F,0xE0,0x0C,0x0F,0xC0,0x0
4,0x07,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,  // ~
};
```