**Micronics Technology H89 ESP32 Interface Status**

**Darrell Pelan**

**22 June 2022**

Contents

[Recent Updates 1](#_Toc106825650)

[Design Status 1](#_Toc106825651)

[Hardware Design 3](#_Toc106825652)

[Ports 3](#_Toc106825653)

[Status 3](#_Toc106825654)

[Commands Under Development 4](#_Toc106825655)

[Notes 5](#_Toc106825656)

[Software 6](#_Toc106825657)

[Startup messages: 6](#_Toc106825658)

[Webpage Processing Messages: 6](#_Toc106825659)

[H89 Esp32 commands 7](#_Toc106825660)

[1. H89 Debug Command Messages: 7](#_Toc106825661)

[2. Get File List 7](#_Toc106825662)

[3. Send File 7](#_Toc106825663)

[4. Get File 7](#_Toc106825664)

[S: Current Status 7](#_Toc106825665)

[X: Reboot Esp32 8](#_Toc106825666)

[Webpage Interface 8](#_Toc106825667)

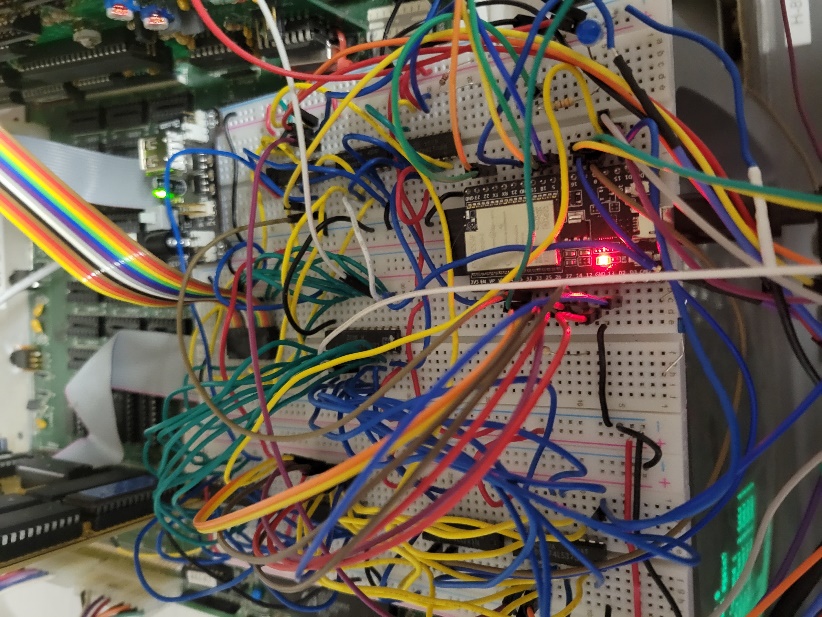
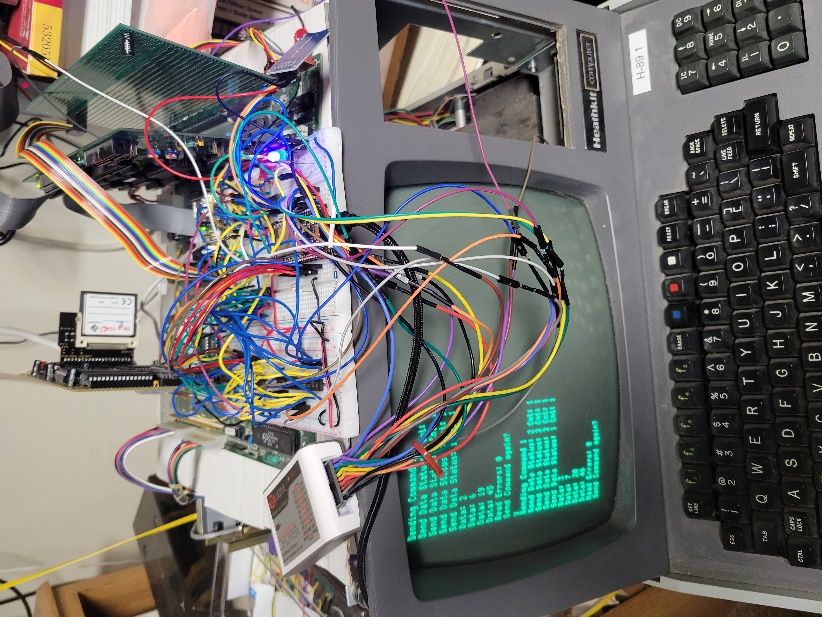
# Recent Updates

* Added commands to transfer files between the esp32 and H89 using a xmodem protocol
* Added H89 command to get file list from the esp32 micro SD card
* Added H89 command to report current esp32 status
* Added H89 command to reboot esp32

# Design Status

The breadboard is currently undergoing software design/ testing. The hardware design is stable. It has a input, output, and status register. The ESP32 uses three interrupts: writes to Port 7E, write to 7C, and a read from 7C. The H-89 determines if the ESP32 is ready for data or data is available by reading port FD.

A picture containing text

Description automatically generated

# Hardware Design

## Ports

The interface is designed to use the three of the H-17 port addresses

|  |  |  |
| --- | --- | --- |
| **H-89 Addr** | **Read Port Function** | **Write Port Function** |
| FC | Read Data | Write Data |
| FD | Read Status | N/A |
| FE | N/A | Tells ESP32 command is coming |

## Status

The ESP32 sends status on two bits using port FD.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Bits** | |  |  |
| **ESP32 Status** | **1** | **0** |  | **Decimal** |
| ESP Ready for Command | 0 | 0 |  | 0 |
| H89 Read Data Ok | 0 | 1 |  | 1 |
| H89 Write Data OK | 1 | 0 |  | 2 |
| ESP Busy | 1 | 1 |  | 3 |

## Commands Under Development

Green highlights are completed.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Command** | **OP Code (HEX)** | **# Bytes** | **Cmd1** | **Cmd2** | **Cmd3** | **Response** |
| Read Status of last operation | 3 | 1 |  |  |  | Send status byte |
| Read sectors | 8 | 4 | Disk|Head | Track|Sector | # sectors | Send # sectors |
| Write Sectors | 0A | 4 | Disk|Head | Track|Sector | # of sectors | Read # sectors, update disk image |
| Seek | 0B | 3 | Disk|Head | Track|Sector |  | Update last operation status |
|  |  |  |  |  |  |  |
| List Files | 10 | 1 |  |  |  | Zero terminated string with list of files on SD card |
| Select disk image | 11 | Varies | Disk # | Zero terminated string with file name | | Assigns file to Disk 1, 2, or 3 |
| Select disk image | 12 | 3 | Disk # | File # from list command | | Assigns file to Disk 1, 2, or 3 |
| Read sectors | 28 | 4 | Disk | LBA MSB | LBA LSB | # sectors | Send # sectors |
| Write Sectors | 2A | 4 | Disk | LBA MSB | LBA LSB | # sectors | Read # sectors, update disk image |
| Seek | 2B | 3 | Disk | LBA MSB | LBA LSB |  |  |
|  |  |  | Disk 3 bits + LBA MSB 5 bits |  |  |  |
| Upload File | 30 | 1 | Zero terminated string with file name | | | Xmodem style transfer, 128 byte blocks |
| Download File | 31 | 1 | Zero terminated string with file name | | | Xmodem style transfer, 128 byte blocks |
| Debug | 1 | 4 |  |  |  |  |
| Reboot | 0 | 1 |  |  |  |  |

**Command byte definitions:**

## Notes

* I’m still working on how to integrate this board with CP/M. The select disk image command is designed to support having the ESP32 act like a disk system. In order for this to be bootable, The H-89 will need a modified MTR-90 EPROM and a default bootable disk image.

# Software

## Startup messages:

Micronics Technology H89 ESP32 interface

This program comes with ABSOLUTELY NO WARRANTY; for details type `L' at the menu.

This is free software, and you are welcome to redistribute it

under certain conditions

WiFi Configuration ...

Loading Configuration ...

Checking Data

ssid key okay

ssid: pelan

Connecting to Wifi:

.WiFi state: Connected

Network Configuration:

----------------------

SSID: pelan

Wifi Status: 3

Wifi Strength: -58 dBm

MAC: 10:97:BD:D4:3C:F4

IP: 192.168.1.9

Subnet: 255.255.255.0

Gateway: 192.168.1.1

DNS 1: 192.168.1.1

DNS 2: 0.0.0.0

DNS 3: 0.0.0.0

Configuring Webserver ...

Elegant OTA Initiated

HTTP Server Has started Sucessfully

To access OTA Update, type

192.168.1.9/update

Hi! I am H89-ESP32, Version 3.4 A 6/6/22

cmdFlag 0

Heap: Free 211404, Min: 211328, Size: 318504, Alloc: 110580

Menu

v: Prints version

b: Reboots system

r: Resets counters

s: SD card test

c: Clears NVM

m: Prints menu

l: Prints license

w: Set up WiFi

## Webpage Processing Messages:

Client:192.168.1.29 / Auth: Failed

Client:192.168.1.29 / Auth: Failed

Client:192.168.1.29 / Auth: Failed

is authenticated via username and password

Client:192.168.1.29 / Auth: Success

Client:192.168.1.29 /favicon.ico

is authenticated via username and password

Client:192.168.1.29 /listfiles Auth: Success

Listing files stored on SD

## H89 Esp32 commands

Text

Description automatically generated

### H89 Debug Command Messages:

The Debug command on the H89 sends four bytes to the ESP32. The ESP adds one and sends back the four bytes.

Interrupt 7E count = 1

Buffer Last 4, Buffer Ptr 0

Cmd Byte 1

Cmd Byte 5

Cmd Byte 18

Cmd Byte 44

Sent 2

Sent 6

Sent 19

Sent 45

Data Out errors: 82618

Interrupt 7C count = 4

### Get File List

Displays files on the Esp32 SD card

### Send File

Sends file using xmodem protocol. File is sent in 128 byte blocks.

### Get File

Gets file from esp32 using xmodem protocol. File is sent in 128 byte blocks. You must include a ‘.’ If there is no filename extension

### S: Current Status

Reports current esp32 status

### X: Reboot Esp32

Restarts the esp32

## Webpage Interface

The network SSIDand password are stored in NVM. If it doesn’t exist, the ESP32 asks for the information, including the user id and password, over the USB interface. You can use the menu to clear NVM.

Text

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated