There is NO ST4 port (I use pulse guide over ascom and never used the SHC) and no indicator leds. These can be added by looking up the necessary info on wiki. I realize pretty soon when I miss tracking :)

My aim was to build an OnStep with the most accurate possible tracking rate available. This is achieved by calculating refraction correction based on the data of the weather sensor and using PPS correction of the base tracking frequency. This combination allows me to use long unguided subs limited by instrument flexure only (combined with extremely precise polar alignment achieved in 10 min with SharpCap)

Axis1, Axis2 TMC5160, focusers LV8729

The TMCs are so called SPI compatible drivers that can control the current and microstep via software. All 4 jumpers under the sockets should be at the low setting. Here is a diagram for your info.

The following is from the FYSETC wiki page:
In order to give you a better understanding of how to use jumpers for different drivers, I have further illustrated the driver socket. The basic definition of the driver pins is the same, the difference lies in the four places of JP6-1, JP6-2, JP6-3 and JP6-4. You can see their role from the schematic.

<table>
<thead>
<tr>
<th>TMCG2130</th>
<th>TMC5160</th>
<th>TMC5161</th>
<th>A4988 etc. Standalone -TMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOSI ----</td>
<td>MS1 ----</td>
<td>MS1 ----</td>
<td>---</td>
</tr>
<tr>
<td>SCK ----</td>
<td>MS2 ----</td>
<td>MS2 ----</td>
<td>---</td>
</tr>
<tr>
<td>CS ----</td>
<td>SPRAED</td>
<td>MS3 ----</td>
<td>---</td>
</tr>
<tr>
<td>MISO ----</td>
<td>PDN ----</td>
<td>RESET</td>
<td>SLEEP</td>
</tr>
<tr>
<td>NC ----</td>
<td>NC ----</td>
<td>NC ----</td>
<td></td>
</tr>
</tbody>
</table>

For TMC2130/5160/5161. You need 4 jumpers to close the JP6, Then the SPI mode will enabled.

With jumpers 1-3 from right you can set the microstep for non SPI drivers.
In case of the LV8729 you should set the microstep level with the jumpers. I suggest using 8. In config set the same microstep as with the jumpers. Below is the code table.

You just need to set the jumpers and insert the drivers in the correct slots with the correct orientation. No soldering is needed up to this point. To connect a motor connector to the relative output sockets you will need the appropriate board side connectors and some soldering at the other end of the cable, so that you can use your favourite connector. I use DSUB connectors for the motors.

### LV8729 motor drive pins description

<table>
<thead>
<tr>
<th>MS1</th>
<th>MS2</th>
<th>MS3</th>
<th>Microstep Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Full Step</td>
</tr>
<tr>
<td>High</td>
<td>Low</td>
<td>Low</td>
<td>1/2 Step</td>
</tr>
<tr>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>1/4 Step</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>1/8 Step</td>
</tr>
<tr>
<td>Low</td>
<td>Low</td>
<td>High</td>
<td>1/16 Step</td>
</tr>
<tr>
<td>High</td>
<td>Low</td>
<td>High</td>
<td>1/32 Step</td>
</tr>
<tr>
<td>Low</td>
<td>High</td>
<td>High</td>
<td>1/64 Step</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>High</td>
<td>1/128 Step</td>
</tr>
</tbody>
</table>

The slots to use for Axis1 and Axis 2 You can see the correct jumper setting.

The slots with the drivers inserted (here focuser1 is inserted only)

I indicated the enable (1) pin on the board so that you can find the correct orientation in case you use different drivers.

The focuser is connected to a DB9 conn with pins 1,2 phase A, 4,5 phase B.
Next let's connect the U-blox NEO 7M GPS unit.

It is a good idea to use a strip connector on the GPS side to be able to remove it while flashing the board. The data flow from the GPS may disable entering into program mode. Alternatively you can use the ATGM336H GPS module, it has a better fix time. The designations are written on the underside of the S6 board and on the silkscreen of the GPS. The EXP1 and EXP2 sockets are at the edge of the board and are clearly indicated with these names.

<table>
<thead>
<tr>
<th>GPS U.blox NEO 7M</th>
<th>FYSETC S6 V2</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCC</td>
<td>EXP1</td>
</tr>
<tr>
<td>GND</td>
<td>EXP1</td>
</tr>
<tr>
<td>TXD</td>
<td>EXP1</td>
</tr>
<tr>
<td>RXD</td>
<td>EXP1</td>
</tr>
<tr>
<td>PPS</td>
<td>EXP2</td>
</tr>
</tbody>
</table>

I suggest to buy a few connectors for the black UART (EXP1, EXP2) sockets on the S6 board because it is very inconvenient to solder directly on the pins.

The LED on the GPS module blinks once a second when a fix was made.

Connecting the WeMos mini Wifi module
I soldered the module directly to the cable but you can use the provided strip connector too. Be careful not to swap orientation, that will kill your module!!

The designations are written on the underside of the S6 board and on the silkscreen of the Wemos.

<table>
<thead>
<tr>
<th>Wemos D1 mini Wifi</th>
<th>FYSETC S6 V2 UART1</th>
</tr>
</thead>
<tbody>
<tr>
<td>5V</td>
<td>VCC</td>
</tr>
<tr>
<td>G</td>
<td>GND</td>
</tr>
<tr>
<td>RX</td>
<td>TX1</td>
</tr>
<tr>
<td>TX</td>
<td>RX1</td>
</tr>
</tbody>
</table>

The wifi module must be flashed before it can be used. When flashing, it must be removed from the S6 board. The details will follow later.

**Connecting the BME280 SPI module**

The designations are written on the underside of the S6 board and on the silkscreen of the BME.

<table>
<thead>
<tr>
<th>BME 280 SPI</th>
<th>FYSETC S6 V2</th>
</tr>
</thead>
<tbody>
<tr>
<td>VCC</td>
<td>TE2 3,3V</td>
</tr>
<tr>
<td>GND</td>
<td>TE2 GND</td>
</tr>
<tr>
<td>SCL</td>
<td>EXP2 SCK</td>
</tr>
<tr>
<td>SDA</td>
<td>EXP2 MOSI</td>
</tr>
<tr>
<td>CSB</td>
<td>EXP2 PC7</td>
</tr>
<tr>
<td>SDO</td>
<td>EXP2 MISO</td>
</tr>
</tbody>
</table>
Programming the FYSETC S6 and the Wifi module

The following preliminary operations are necessary:

Install the Arduino IDE (I used ver 1.8.5 for this project)
IN the Arduino IDE:
Install the necessary driver for the S6 card (Files/Preferences/Additional Boards...)
https://github.com/stm32duino/BoardManagerFiles/raw/master/STM32/package_stm_index.json
Install the necessary driver for the ESP8266 wifi module (Files/Preferences/Additional Boards...)

Add the following libraries (Sketch/Include library/Add ZIP library)
BME280, Tinygps++(http://arduiniana.org/libraries/tinygpsplus)

Install the STM32CUBEPROG from the link on the FYSETC S6 wiki page. Install the STM32Duino Board Manager. version 1.9.0 only

Download the OnStep firmware https://onstep.groups.io/g/main/wiki/3915
Unpack it into a folder called “Onstep” I used ver 4.24

The only part of the code that you have to customize is the “config.h” file. This is where you find it after opening the "OnStep.ino" file with the Arduino IDE. (wifi.ino in case of the wifi module) Alternatively you can open the config files directly in your favourite editor.

You can see my config.h file at the end of this pdf for reference. Naturally, your steps/degree values will have to be calculated with the configurator.
Important: On Windows, the STM32Duino package uses a script to call the STM32CubeProgrammer in command-line mode to perform the actual upload. This script contains an incorrect address for the code segment start address (its probably correct for some boards but not the FYSETC S6). Currently the only way to change this to the correct value is to edit the script according to these instructions:

- Locate the file `stm32CubeProg.bat`. It should be at `~\AppData\Local\Arduino15\Packages\STM32\Tools\STM32tools\1.4.0\tools\win\STM32CubeProgrammer\stm32CubeProg.bat` but if the location changes in the future due to package updates you may need to search for it starting at `~\AppData\Local\Arduino15`.
- Open the file in your favorite editor
- Change the value on the 'SET ADDRESS' line from 0x8000000 to 0x8010000
- Save and exit

Once all the above has been done, in (IDE) Tools/Boards select 3D printer boards and select exactly the following options, otherwise some things will not work (e.g. USB port)

- Board -> 3D printer boards
- Optimize -> Fastest (-O3)
- Board part number -> FYSETC_S6
- C Runtime Library -> Newlib Nano + Float Printf/Scanf
- USB Speed (if available) -> Low/Full Speed
- USB Support (if available) -> CDC (generic 'Serial' supercede U(S)ART)
- U(S)ART Support -> Enabled (generic 'Serial')
- Upload Method -> STM32CubeProgrammer (DFU)

Compile the firmware to see if there is no error. Now you need to get the S6 board in program mode.
Put the 5V power jumper near the USB port to USB (so that no external power supply is needed). Alternatively, you can leave it in DC and have 12VDC connected while flashing.

First power off the board, then jumper the Boot0 pin to the 3.3v pin, then connect the USB to the board and your computer, then press the reset button. The card will enter DFU mode. Now you can start upload in the IDE.

If the IDE cannot connect it means that your card did not enter DFU mode, Try to repeat the power cycle and the reset.

Try to push the reset button within 2 sec after applying power to avoid the GPS unit interference. Alternatively, remove the GPS unit while flashing the S6 board.

**REMEMBER to remove the boot jumper if you finish uploading or it will enter DFU mode again. Position the power jumper to DC.**

Now program the Wemos mini wifi module with the Arduino IDE.

Remove the module from the S6 card and connect via (micro) usb to the PC.
If nor done yet, install the driver for the card in the Card Manager, **version 2.4.2 is needed.** You find the firmware in the downloaded Onstep firmware in a subfolder: addons/wifiEdit the wifi config.h (weather ON, baud rate as set in the S6 card firmware, I use 57600) Before starting the compilation, select generic ESP8266 (or WeMos D1 R1) in Tools/card. If the firmware compiles, you can upload it. Once done, remove the module from the USB cable and connect it to the S6 board. The status LED blinks for some seconds, then it is stable when connection has been established.

If you want to connect encoder to the WeMos WiFi module, it is important to disable Track Autostart in the config file, otherwise the module will not boot at startup.
This is all, your controller is functional now.

I owe spacial thanks to Emanuele De Giorgio and Khalid Baheyeldin for their valuable help.

The FYSETC S6 wiki page
https://onstep.groups.io/g/main/wiki/21159

This is my main config settings that compile without errors and work for the above presented hardware. You will need to change the mount/motor and focuser specific parameters only. I have omitted the last part as there is no change there at all.

// Configuration for OnStep

/*
 * For more information on setting OnStep up see http://www.stellarjourney.com/index.php?r=site/equipment_onstep
 * and join the OnStep Groups.io at https://groups.io/g/onstep
 * *** Read the compiler warnings and errors, they are there to help guard against invalid configurations ***
 */

// ADJUST THE FOLLOWING TO CONFIGURE YOUR CONTROLLER FEATURES

// PINMAP ------------------------------------------------- see https://onstep.groups.io/g/main/wiki/Configuration-Controller#PINMAP
// *** See the matching Pins.xxx.h file for your setup (found by looking in src/pinmaps/Models.h) with detailed information ***
// *** to be sure it matches your wiring. *** USE AT YOUR OWN RISK ***                                           
#define PINMAP FYSETC_S6_2 //    OFF, Choose from: MksGenL2, MiniPCB2, MaxPCB2, MaxESP3, CNC3,
//         MaxSTM3, FYSETC_S6_2, etc. Other boards and more info. in Constants.h

// SERIAL PORT COMMAND CHANNELS --------------------------- see https://onstep.groups.io/g/main/wiki/Configuration-Controller#SERIAL
#define SERIAL_A_BAUD_DEFAULT 57600 //   9600, n. Where n=9600,19200,57600,115200 (common baud rates.) 
Infreq 
#define SERIAL_B_BAUD_DEFAULT 57600 //   9600, n. See (src/HAL/) for your MCU Serial port # etc.                        Option
#define SERIAL_B_ESP_FLASHING OFF //    OFF, ON Upload ESP8266 WiFi firmware through SERIAL_B with :ESPFLASH# cmd.    Option 
#define SERIAL_C_BAUD_DEFAULT OFF //    OFF, ON for ESP32 Bluetooth.                                               Option
#define SERIAL_C_BLUETOOTH_NAME "OnStep" // "On..", Bluetooth device name for ESP32.                                         Option

// MOUNT TYPE ----------------------------------------- see https://onstep.groups.io/g/main/wiki/Configuration-Controller#MOUNT
#define MOUNT_TYPE GEM //    GEM, GEM for German Equatorial, FORK for Equatorial Fork, or ALTAZM          <-
Req'd
// Dobsonian etc. mounts. GEM Eq mounts perform meridian flips.

// USER FEEDBACK -------------------------------------- see https://onstep.groups.io/g/main/wiki/Configuration-Controller#USER
#define LED_STATUS OFF //  ON, Blinks w/sidereal tracking otherwise steady on indicates activity.     Option 
#define LED_STATUS2 OFF // OFF, n. Where n=0..255 (0..100%) activates feature sets default brightness. Option
#define BUZZER OFF // OFF, n. Where n=0..6000 (Hz freq.) for piezo speaker. ON for buzzer.       Option
#define BUZZER_STATE_DEFAULT OFF // OFF, ON Start with piezo buzzer/speaker enabled.              Option

// TIME AND LOCATION ----------------------------------- see https://onstep.groups.io/g/main/wiki/Configuration-Controller#TLS
#define TIME_LOCATION_SOURCE GPS //  OFF, DS3231 (I2c, DS3234 (Spi,) TEENSY (T3.2 internal,) or GPS source. Option
#define SERIAL_GPS Serial3

GTD 09 2021 10 OnStep FYSETC S6 build
#define SerialGPSBaud 9600  // Provides Date/Time, and if available, PPS & Lat/Long also.

// SENSORS ----------------------------------------------- see https://onstep.groups.io/g/main/wiki/Configuration-Controller#SENSORS
// * = also supports ON_PULLUP or ON_PULLDOWN to activate MCU internal resistors if present.
#define WEATHER BME280_SPI // OFF, BME280 (I2C 0x77,) BME280_0x76, BME280_SPI (see pinmap for CS.)
  Option
    // BMP280 (I2C 0x77,) BMP280_0x76, BMP280_SPI (see pinmap for CS.)
    // BME280 or BMP280 for temperature, pressure. BME280 for humidity also.

#define TELESCOPE_TEMPERATURE OFF // OFF, DS1820, n. Where n is the DS1820 s/n for focuser temperature.
    Adjust

#define HOMESENSE OFF // OFF, ON*. Automatically detect and use home switches. For GEM mode only.
  Option
#define HOMESENSE_STATE_AXIS1 HIGH // HIGH, State when clockwise of home position, as seen from front. Rev.
  w/LOW. Adjust
#define HOMESENSE_STATE_AXIS2 HIGH // HIGH, State when clockwise of home position, as seen from above. Rev.
  w/LOW. Adjust
  // Signal state reverses when travel moves ccw past the home position.

#define LIMITSENSE OFF // OFF, ON* limit sense switch close to Gnd stops gotos and/or tracking. Option
#define LIMITSENSE_STATE LOW // LOW, For NO (normally open) switches, HIGH for NC (normally closed.)
  Adjust

#define PECSENSE OFF // OFF, ON*, n, sense digital OR n=0 to 1023 (0 to 3.3V or 5V) analog threshold. Option
#define PECSENSE_STATE HIGH // HIGH, Senses the PEC signal rising edge or use LOW for falling edge. Adjust
  // Ignored in ALTAZM mode.

#define PPSSENSE OFF // OFF, ON* enables PPS (pulse per second,) senses signal rising edge. Option
  // Better tracking accuracy especially for Mega2560's w/ceramic resonator.

// ST4 INTERFACE --------------------------------------------- see https://onstep.groups.io/g/main/wiki/Configuration-Controller#ST4
// *** It is up to you to verify the interface meets the electrical specifications of any connected device, use at your own risk ***
#define ST4INTERFACE OFF // OFF, ON, ON_PULLUP enables interface. <= 1X guides unless hand control mode. Option

#define ST4_HAND_CONTROL OFF // OFF, ON for hand controller special features and SHC support. Option
  // Hold [E]+[W] btns >2s: Guide rate [E]- [W]+ [N] trk on/off [S] sync
  // Hold [N]+[S] btns >2s: Usr cat item [E]- [W]+ [N] goto [S] and on/off
#define ST4_HAND_CONTROL_FOCUSER OFF // OFF, ON alternate to above: Focuser move [E]f1 [W]f2 [N]- [S]+
  Option

// GUIDING BEHAVIOUR ------------------------------------------ see https://onstep.groups.io/g/main/wiki/Configuration-Mount#GUIDING
#define GUIDE_TIMELIMIT 0 // 0, No guide time limit. Or n. Where n=1..120 second time limit guard. Adjust
#define GUIDE_DISABLE_BACKLASH OFF // OFF, Disable backlash takeup during guiding at <= 1X
  Option

// TRACKING BEHAVIOUR ---------------------------------------- see https://onstep.groups.io/g/main/wiki/Configuration-Mount#TRACKING
#define TRACK_AUTOSTART ON // OFF, ON Start with tracking enabled. Option
#define TRACKREFRACTION_RATE_DEFAULT ON // OFF, ON Start w/atmospheric refract. compensation (RA axis/Eq mounts only.) Option
#define TRACK_BACKLASH_RATE 25 // 25, n. Where n=2.50 (x sidereal rate) during backlash takeup. Option
  // Too fast motors stall/gears slam or too slow and sluggish in backlash.

// SLEWING BEHAVIOUR ----------------------------------------- see https://onstep.groups.io/g/main/wiki/Configuration-Mount#SLEWING
#define SLEW_RATE_BASE_DESIRED 2.0 // 1.0, n. Desired slew rate in deg/sec. Adjustable at run-time from Req'd
  // 1/2 to 2x this rate, and as MCU performace considerations require.
#define SLEW_RATE_MEMORY ON // OFF, ON Remembers rates set across power cycles. Option
#define SLEW_ACCELERATION_DIST 5.0 // 5.0, n. (degrees.) Approx. distance for acceleration (and deacceleration.) Adjust
#define SLEW_RAPID_STOP_DIST 2.5 // 2.0, n. (degrees.) Approx. distance required to stop when a slew is aborted or a limit is exceeded.

GTD 09 2021
OnStep FYSETC S6 build
// PIER SIDE BEHAVIOUR ---------------------------------------- see https://onstep.groups.io/g/main/wiki/Configuration-Mount#SYNCING
#define MFLIP_SKIP_HOME          OFF // OFF, ON Goto directly to the destination without visiting home position. Option
#define MFLIP_PAUSE_HOME_MEMORY   OFF // OFF, ON Remember meridian flip pause at home setting across power cycles. Option
#define MFLIP_AUTOMATIC_MEMORY    ON // OFF, ON Remember automatic meridian flip setting across power cycles. Option
#define PIER_SIDE_SYNC_CHANGE_SIDES ON // OFF, ON Allows sync to change pier side, for GEM mounts. Option
#define PIER_SIDE_PREFERRED_DEFAULT BEST // BEST, Stays on current side if possible. EAST or WEST switch if possible. Option

// PARKING BEHAVIOUR ------------------------------------------ see https://onstep.groups.io/g/main/wiki/Configuration-Mount#PARKING
#define STRICT_PARKING           OFF // OFF, ON Un-parking is only allowed if successfully parked. Option

// MOTION CONTROL --------------------------------------------- see https://onstep.groups.io/g/main/wiki/Configuration-Mount#MOTION
#define STEP_WAVE_FORM            SQUARE // SQUARE, PULSE Step signal wave form faster rates. SQUARE best signal integrity. Adjust

// Stepper driver models (also see ~/OnStep/src/sd_drivers/Models.h for additional infrequently used models and more info.):
// A4988, DRV8825, LV8729, S109, SSS TMC2209*, TMC2130* **, and TMC5160* ***
// * = add _QUIET (stealthChop tracking) for example "TMC2130_QUIET"
// ** = SSS TMC2130 if you choose to set stepper driver current (in mA) set Vref pot. 2.5V instead of by motor current as usual.
// *** = SSS TMC5160 you must set stepper driver current (in mA) w/ #define AXISn_TMC_IRUN (I霍LD, etc.)
// Settings for driver Microsteps, IRUN, Reverse, Limit Min, and Limit Max are stored in NV (EEPROM.) These runtime settings
// can be changed (or reverted to the defaults below) from the SmartWebServer's Config webpage. If runtime axis settings are enabled changes to these settings below may be ignored as runtime settings from NV (EEPROM) are used instead.

// AXIS1 RA/AZM
// see https://onstep.groups.io/g/main/wiki/Configuration-Mount#AXIS1
#define AXIS1_STEPS_PER_DEGREE    34312 // 12800, n. Number of steps per degree: -=Req'd
    // n = (stepper_steps * micro_steps * overall_gear_reduction)/360.0
#define AXIS1_STEPS_PER_WORMROT   0 // 0, n. Number steps per worm rotation (PEC Eq mode only, 0 disables PEC.) -=Req'd
    // n = (AXIS1_STEPS_PER_DEGREE*360)/reduction_final_stage
#define AXIS1_DRIVER_MODEL        TMC5160 // OFF, (See above.) Stepper driver model. -=Often
#define AXIS1_DRIVER_MICROSTEPS   128 // OFF, n. Microstep mode when tracking. -=Often
#define AXIS1_DRIVER_MICROSTEPS_GOTO 8 // OFF, n. Microstep mode used during gotos. Option
#define AXIS1_DRIVER_IHOLD        300 // OFF, n, (mA.) Current during standstill. OFF uses IRUN/2.0 Option
#define AXIS1_DRIVER_IRUN         900 // OFF, n, (mA.) Current during tracking, appropriate for stepper/driver/etc. Option
#define AXIS1_DRIVER_IGOTO        1400 // OFF, n, (mA.) Current during slews. OFF uses same as IRUN. Option
#define AXIS1_DRIVER_REVERSE      OFF // OFF, ON Reverses movement direction, or reverse wiring instead to correct. -=Often
#define AXIS1_DRIVER_STATUS       OFF // OFF, TMC_SPI, HIGH, or LOW. Polling for driver status info/fault detection. Option
#define AXIS1_LIMIT_MIN           -180 // -180, n. Where n= -90..-270 (degrees.) Minimum "Hour Angle" for Eq modes. Adjust
    // n. Where n= -180..-360 (degrees.) Minimum Azimuth for AltAzm mode.
#define AXIS1_LIMIT_MAX           180 // 180, n. Where n=  90.. 270 (degrees.) Maximum "Hour Angle" for Eq modes. Adjust
    // n. Where n= 180.. 360 (degrees.) Maximum Azimuth for AltAzm mode.

// AXIS2 DEC/ALT
// see https://onstep.groups.io/g/main/wiki/Configuration-Mount#AXIS2
#define AXIS2_STEPS_PER_DEGREE    51560 // 12800, n. Number of steps per degree: -=Req'd
    // n = (stepper_steps * micro_steps * overall_gear_reduction)/360.0
#define AXIS2_DRIVER_MODEL        TMC5160 // OFF, (See above.) Stepper driver model. -=Often
#define AXIS2_DRIVER_MICROSTEPS   128 // OFF, n. Microstep mode when tracking. -=Often
#define AXIS2_DRIVER_MICROSTEPS_GOTO 8 // OFF, n. Microstep mode used during gotos. Option
#define AXIS2_DRIVER_IHOLD        300 // OFF, n, (mA.) Current during standstill. OFF uses IRUN/2.0 Option
GTD 09 2021
12
OnStep FYSETC S6 build
```c
#define AXIS2_DRIVER_IRUN 700 // OFF, n, (mA.) Current during tracking, appropriate for stepper/driver/etc. Option
#define AXIS2_DRIVER_IGOTO 1400 // OFF, n, (mA.) Current during slews. OFF uses same as IRUN. Option
#define AXIS2_DRIVER_POWER_DOWN OFF // OFF, ON Powers off 10sec after movement stops or 10min after last<=1x guide. Option
#define AXIS2_DRIVER_REVERSE OFF // OFF, ON Reverses movement direction, or reverse wiring instead to correct. Often
#define AXIS2_DRIVER_STATUS OFF // OFF, TMC_SPI, HIGH, or LOW. Polling for driver status info/fault detection. Option
#define AXIS2_DRIVER_TANGENT_ARM OFF // OFF, ON limit range below. Set cntr w/[Reset Home] Return cntr w/[Find Home] Infreq
#define AXIS2_LIMIT_MIN -90 // -90, n. Where n=-90..0 (degrees.) Minimum allowed declination. Infreq
#define AXIS2_LIMIT_MAX 90 // 90, n. Where n=0..90 (degrees.) Maximum allowed declination. Infreq

// AXIS3 ROTATOR
// see https://onstep.groups.io/g/main/wiki/Configuration-Rotator-and-Focusers#AXIS3
#define ROTATOR OFF // OFF, ON to enable the rotator (or de-rotator for ALTAZM mounts.) Option
#define AXIS3_STEPS_PER_DEGREE 64.0 // 64.0, n. Number of steps per degree for rotator/de-rotator. Adjust
#define AXIS3_SLEW_RATE_DESIRED 1.0 // 1.0, n, (degrees/second) Maximum speed depends on processor. Adjust
#define AXIS3_DRIVER_MODEL OFF // OFF, TMC2130, TMC5160. Leave OFF for all drivers models except these. Option
#define AXIS3_DRIVER_MICROSTEPS OFF // OFF, n. Microstep mode when tracking. For TMC2130, TMC5160. Option
#define AXIS3_DRIVER_IHOLD OFF // OFF, n, (mA.) Current standstill. OFF uses IRUN/2.0. " Option
#define AXIS3_DRIVER_IRUN OFF // OFF, n, (mA.) Current tracking, appropriate for stepper/driver/etc. " Option
#define AXIS3_DRIVER_POWER_DOWN OFF // OFF, ON Powers off the motor at stand-still. Option
#define AXIS3_DRIVER_REVERSE OFF // OFF, ON Reverses movement direction, or reverse wiring instead to correct. Option
#define AXIS3_LIMIT_MIN -180 // -180, n. Where n=-360..0 (degrees.) Minimum allowed rotator angle. Infreq
#define AXIS3_LIMIT_MAX 180 // 180, n. Where n=0..360 (degrees.) Maximum allowed rotator angle. Infreq

// AXIS4 FOCUSER 1
// see https://onstep.groups.io/g/main/wiki/Configuration-Rotator-and-Focusers#AXIS4
#define FOCUSER1 ON // OFF, ON to enable this focuser. Option
#define AXIS4_STEPS_PER_MICRON 12 // 0.5, n. Steps per micrometer. (at 8 micro)Figure this out by testing or other means. Adjust
#define AXIS4_SLEW_RATE_DESIRED 1000 // 500, n, Where n=200..5000 (um/s.) Max microns/second. In DC mode, max pwr % Adjust
#define AXIS4_DRIVER_MODEL OFF // OFF, OFF, TMC2130, TMC5160. Leave OFF for all drivers models except these. Option
#define AXIS4_DRIVER_MICROSTEPS 8 // OFF, n. Microstep mode when tracking. For TMC2130, TMC5160. Option
#define AXIS4_DRIVER_IHOLD OFF // OFF, n, (mA.) Current standstill. OFF uses IRUN/2.0. " Option
#define AXIS4_DRIVER_IRUN OFF // OFF, n, (mA.) Current tracking, appropriate for stepper/driver/etc. " Option
#define AXIS4_DRIVER_POWER_DOWN OFF // OFF, ON Powers off the motor at stand-still. Option
#define AXIS4_DRIVER_REVERSE OFF // OFF, ON Reverses movement direction, or reverse wiring instead to correct. Option
#define AXIS4_LIMIT_MIN_RATE 100 // 50, n. Where n=1..1000 (um/s.) Minimum microns/second. In DC mode, min pwr % Adjust
#define AXIS4_LIMIT_MIN 100 // 100, n. Where n=0.500 (millimeters.) Minimum allowed position. Adjust
#define AXIS4_LIMIT_MAX 100 // 50, n. Where n=0..500 (millimeters.) Maximum allowed position. Adjust

// AXIS5 FOCUSER 2
// see https://onstep.groups.io/g/main/wiki/Configuration-Rotator-and-Focusers#AXIS5
#define FOCUSER2 ON // OFF, ON to enable this focuser. Option
#define AXIS5_STEPS_PER_MICRON 12 // 0.5, n. Steps per micrometer. Figure this out by testing or other means. Adjust
#define AXIS5_SLEW_RATE_DESIRED 1000 // 500, n, Where n=200..5000 (um/s.) Max microns/second. In DC mode, max pwr % Adjust
```
This is the configuration of the WeMos Mini wifi module I used

// Configuration for OnStep WiFi Add-on

/
* For more information on setting this addon up see https://onstep.groups.io/g/main/wiki/7119
* and join the OnStep Groups.io at https://groups.io/g/onstep
* ** Read the compiler warnings and errors, they are there to help guard against invalid configurations **
*/

// ADJUST THE FOLLOWING TO CONFIGURE YOUR ADD-ON'S FEATURES

// SERIAL PORTS
#define SERIAL_BAUD_DEFAULT 57600 // 9600, Common baud rates for these parameters are 9600,19200,57600,115200. Infreq
#define SERIAL_BAUD 57600 // 57600, Automatically uses 19200 if talking to a Mega2560 OnStep. Infreq
// If establishing a link to OnStep was unsuccessful*** the ESP8266 may
// retain prior settings perhaps an SSID from factory defaults, for example.
#define SERIAL_SWAP AUTO // AUTO, Automatic check both, ON for swapped port or OFF for default port only. Infreq

// USER FEEDBACK
#define LED_STATUS 2 // 2, GPIO LED pin WeMos D1 Mini. Flashes connecting then steady on connected. Infreq
#define DISPLAY_LANGUAGE L_en // L_en, English. Specify language with two letter country code, if supported. Adjust
#define DISPLAY_WEATHER ON // OFF, Ambient conditions in locale default units. Option
#define DISPLAY_INTERNAL_TEMPERATURE OFF // OFF, Internal MCU temp. in locale default units. Option
#define DISPLAY_WIFI_SIGNAL_STRENGTH ON // OFF, Wireless signal strength reported via web interface. OFF otherwise. Option
#define DISPLAY_SPECIAL_CHARS ON // ON, For standard ASCII special symbols (compatibility.) Infreq
#define DISPLAY_ADVANCED_CHARS ON // OFF, For standard "RA/Dec" instead of symbols. Infreq

GTD 09 2021
#define DISPLAY_HIGH_PRECISION_COORDS OFF // OFF, ON for high precision coordinate display on status page.

// COMMAND CHANNELS----------------------------------------------------------------------------------------
#define STANDARD_COMMAND_CHANNEL ON // ON, Enable standard cmd channel port 9999 use w/Android App & ASCOM driver. Infreq
#define PERSISTENT_COMMAND_CHANNEL ON // ON, Enable persistent cmd channel port 9998 use w/INDI? & Stellarium Mobile. Infreq
   // Experimental, possibly causes problems w/standard cmd channel if enabled.

// ENCODER SUPPORT----------------------------------------------------------------------------------------
#define ENC_AUTO_SYNC_DEFAULT ON // ON, Automatically sync Encoders to OnStep.                        Option
#define AXIS1_ENC OFF // OFF, CWCCW, AB, BC_BISSC. RA/Azm Axis (A/CW/MA) & (B/CCW/SLO.) Option
#define AXIS1_ENC_REVERSE OFF // OFF, ON to reverse the count direction.                              Adjust
#define AXIS1_ENC_TICKS_DEG 22.22222 // 22.222, n, (ticks/degree.) Encoder ticks per degree.        Adjust
#define AXIS1_ENC_DIFF_LIMIT_TO 300 // 300, n, (arcsec.) Minimum diff. between encoder/OnStep for sync. to OnStep. Adjust
#define AXIS1_ENC_DIFF_LIMIT_FROM OFF // OFF, n, (arcsec.) Maximum diff. between encoder/OnStep for sync. from OnStep. Adjust
   // For absolute encoders, leave off when setting Zero, then enable.

#define AXIS2_ENC OFF // OFF, CWCCW, AB, BC_BISSC. Dec/Alt Axis (A/CW/MA) & (B/CCW/SLO.) Option
#define AXIS2_ENC_REVERSE OFF // OFF, ON to reverse the count direction.                              Option
#define AXIS2_ENC_TICKS_DEG 22.22222 // 22.222, n, (ticks/degree.) Encoder ticks per degree.        Adjust
#define AXIS2_ENC_DIFF_LIMIT_TO 300 // 300, n, (arcsec.) Minimum diff. between encoder/OnStep for sync. to OnStep. Adjust
#define AXIS2_ENC_DIFF_LIMIT_FROM OFF // OFF, n, (arcsec.) Maximum diff. between encoder/OnStep for sync. from OnStep. Adjust
   // For absolute encoders, leave off when setting Zero, then enable.

// ENCODER RATE CONTROL                                                                              
#define AXIS1_ENC_RATE_CONTROL OFF // OFF, ON Rate control for RA high resolution encoder. EQ mounts only. Infreq
#define AXIS1_ENC_INTPOL_COS OFF // OFF, ON enables cosine compensation feature.                       Infreq
#define AXIS1_ENC_RATE_AUTO OFF // OFF, n, (Worm period in seconds.) Adjusts avg encoder pulse rate to account. Option
   // For skew in the average guide rate over the last worm period.                                     Option
#define AXIS1_ENC_BIN_AVG OFF // OFF, n, (Number of bins.) Enables binned rolling average feature.     Option

// THAT'S IT FOR USER CONFIGURATION!
// -------------------------------------------------------------------------------

GTD 09 2021
OnStep FYSETC S6 build