Version 1

el Comandante http://guerilla.bplaced.net

ESP32 Robot 1 - Arduino Software

Preface:

The goal was to control a 30 years old RC car toy with 2 motors remotly using the wireless ESP-NOW protocol and using the DRV8833 motor driver, no sensoric included.
The complete software bundle for the Arduino IDE contains:

- The Arduino transmitter software for the WT32-SC01 device
- The Arduino receiver software for the LILYGO TTGO-T-Display device
- This pdf manual

About the ESP32:

GPIO 0, GPIO 2, GPIO 4, GPIO 5, GPIO 12 and GPIO 15 are strapping pins that bring the ESP32 into bootloader or flash mode while booting. Connecting something to it during boot which pull the pins into the HIGH or LOW state can lead to unexpected behavior.
The ADC2_CHx pins cannot be used as analog inputs when WiFi is active.
GPIO 34 - 39 are inputs only, no internal pull-up or pull-down resistor available on these pins.
Using GPIOs as an input must be enabled with INPUT_PULLDOWN when using the ESP32.
This are the **default** esp32 pins for the HSPI, the VSPI and the I2C interface:

HSPI_MISO	HSPI_MOSI	HSPI_CLK	HSPI_CS	VSPI_MISO	VSPI_MOSI	VSPI_CLK	VSPI_CS	I2C_SDA	I2C_SCL
GPIO 12	GPIO 13	GPIO 14	GPIO 15	GPIO 19	GPIO 23	GPIO 18	GPIO 5	GPIO 21	GPIO 22

About the WT32-SC01 ESP32 Development Board:

ESP32-WROVER-B, 4 MB SPI Flash and 8 MB PSRAM (16 MB version available on Alibaba)
Dual-Core XtensaRO 32-Bit LX6 MCU, 240 Mhz
3.5" 320 x 480 TFT (ST7796), capacitive 2-Touch (FT6336U)
USB-C, WiFi 2.4 GHz WLAN 802.11 b/g/n, Bluetooth 4.2 BR/EDR- and BLE-Standards, 5V DC
Issues: Reading TFT pixel using HSPI with the TFT_eSPI or the LovyanGFX library isn't working.

TFT_MISO	TFT_MOSI	TFT_SCLK	TFT_CS	TFT_DC	TFT_RST	TFT_BL	I2C_SDA	I2C_SCL
GPIO 12	GPIO 13	GPIO 14	GPIO 15	GPIO 21	GPIO 22	GPIO 23	GPIO 18	GPIO 19

About the LILYGO TTGO-T-Display ESP32 Development Board:

ESP32-WROVER-B, 520 KB SRAM, 4 MB or 16 MB SPI Flash
Dual-Core XtensaRO 32-Bit LX6 MCU, 240 Mhz, CH9102F chipset
1.14" 240 x 135 IPS TFT (ST7789V)
USB-C, WiFi 2.4 GHz WLAN 802.11 b/g/n, Bluetooth 4.2 BR/EDR- and BLE-Standards, 5V DC

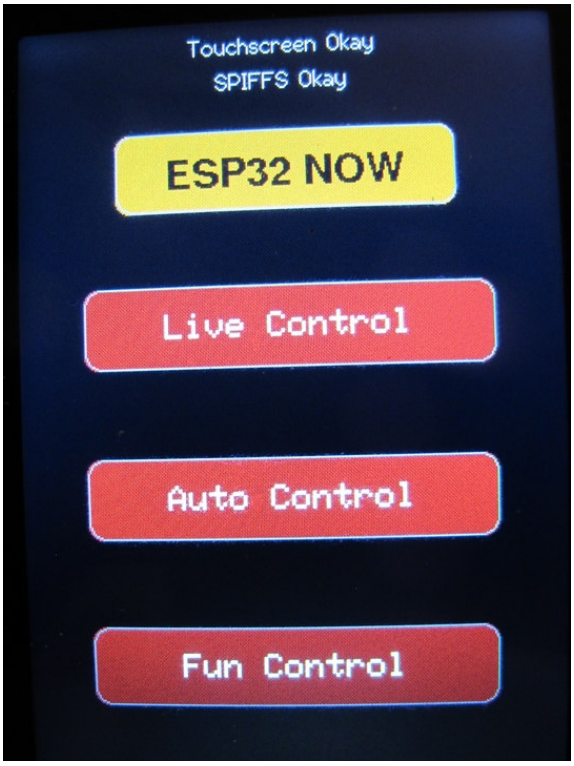
TFT_MISO	TFT_MOSI	TFT_SCLK	TFT_CS	TFT_DC	TFT_RST	TFT_BL	I2C_SDA	I2C_SCL
N/A	GPIO 19	GPIO 18	GPIO 5	GPIO 16	GPIO 23	GPIO 4	GPIO 21	GPIO 22

ESP32 Robot 1 - Arduino Software

ESP32-NOW, WT32-SC01 (Transmitter) – V1

Documentations and Libraries:

Overview ESP32 - https://socialcompare.com/en/comparison/esp8266-vs-esp32-vs-esp32-s2
Espressif Docs - https://www.espressif.com/en/support/documents/technical-documents
WT32-SC01 - http://myosuploads3.banggood.com/products/20220111/20220111020446WT32-SC01DataSheetV3.3.pdf
TFT ST7796 - https://cdn.hackaday.io/files/1625926956336128/ST7796S.pdf
Touch FT6336U - https://focuslcds.com/content/FT6236.pdf
TFT_eSPI library - https://github.com/Bodmer/TFT_eSPI
ST7796 library - included in the TFT_eSPI library
FT6336U library - https://github.com/adafruit/Adafruit_FT6206_Library - **Modify Wire.begin() to Wire.begin(18, 19)**



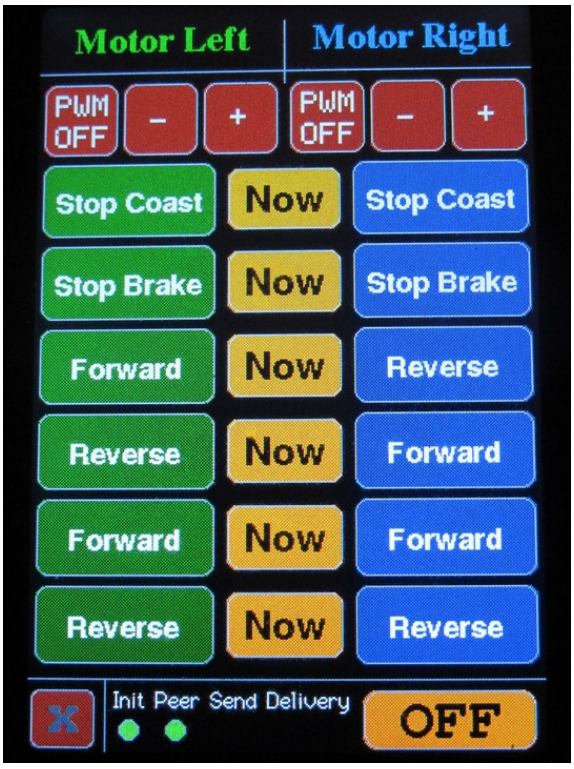
Main Menu:

On startup the program inits the touchscreen and the filesystem (could take some seconds). The result will be displayed on top of the screen. To control the robot three modes are available.

Live Control - Send commands to the left, to the right or to both motors. The command will be executed until sending a new command.

Auto Control - The user can program a sequence of steps. After sending the sequence the motors follow the received instructions. In the Auto options menu a sequence can be loaded or saved.

Fun Control - The WT32-SC01 act like a remote control for a car toy. As long as a button is touched the robot executes the command.



Live Control:

The area at the bottom of the screen signals the Wifi status, success (Green) or error (Red)

Touch a green button to control the left motor

Touch a blue button to control the right motor

Touch a “Now“ button to control both motors

Touch a red button to increase or decrease the speed (the PWM value) of every motor. PWM can also switched off for the maximum speed

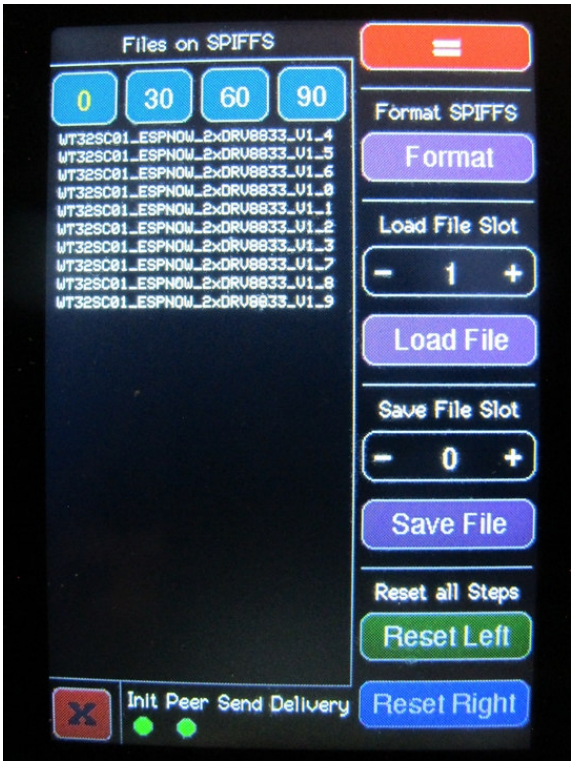
The “OFF” button emergency stops both motors.

The red button with the cross reboots the WT32-SC01

Auto Control:

= – Display Auto Control Options Menu
Motor selection – Left motor | Right motor
Step Number – Value (0 - 255)
Start Time in s – Value in 100 ms steps up to 10000.0 s
Direction – Off | Forward | Coast | Brake | Reverse
Control Mode – Off | FIX | PWM
PWM – Value (0 - 255)
ADD – Adds a step direct after the current step. Upper steps moved 1 step upwards. The highest step is lost.
DEL – Deletes the current step. Upper steps moved 1 step downwards. The last step is new and set to zero.
Sync Step – Copy the current step of the current motor to the same step number of the other motor
Now – Transmit step sequence to the target device
The red button with the cross reboots the WT32-SC01

The area at the bottom of the screen signals the WiFi status, success (Green) or error (Red).
Touch the vertical area below „Motor Right“ to set the „Start Time in s“ value multiplicator.
Touch the vertical area below „Motor Left“ to adjust the Timeline maximum.
Touch the vertical area below „Time“ to set the Timeline Base.
The lines of the kindly visualized steps should not cross each other!
Invalid steps are ignored by the receiver software, i.e. if step 2 time is below step 1 time.
The maximum steps to transmit is the index of the last step where the start time is not 0.

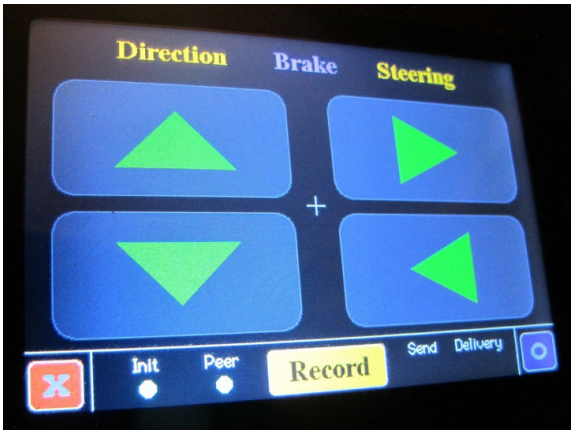


Auto Control Options:

The area at the bottom of the screen signals the WiFi status, success (Green) or error (Red).

= – Display Auto Control Menu
0 – Display Files 0 - 29
30 – Display Files 30 - 59
60 – Display Files 60 - 89
90 – Display Files 90 - 119
Format – Format SPIFFS
Load File Slot – Value (0 - 19)
Load File – Load File using the selected slot number
Save File Slot – Value (1 - 19)
Save File – Save File using the selected slot number
Reset Left – Reset all left motor steps
Reset Right – Reset all right motor steps
The red button with the cross reboots the WT32-SC01

The maximum steps saved is the index of the last step where the start time is not 0.
File Slot 0 is reserved for the Fun Control Mode recording option.
In the Auto Control mode some buttons (touch areas) for critical functions must be touched longtime (>= 1500ms).



Fun Control:

The area at the bottom of the screen signals the WiFi status, success (Green) or error (Red)
Touch the left buttons for the direction
Touch the right buttons for the steering
The “Record” button starts / stops recording
The record will only be saved after stopping a record
The purple button change the steering mode
The red button with the cross reboots the WT32-SC01

In Fun Control mode the Reboot, the Record and the Steering Mode buttons (touch areas) at the bottom of the screen must be touched longtime (>= 1000ms).
PWM isn‘t implemented in Fun Control.
Unfortunately the touch screen is not so exact as it should be!
Maybe, in a future release, it will be optimized. If possible.
If the maximum allowed steps are reached the record stops and saves automaticly.
On recording the button is colored red.
On success while saving the text „Record“ remains black, on error the text becomes red.
Starting a new record the old recorded file „WT32SC01_ESPNOW_DRV8833_V1_0“ will be overwritten. To keep a saved record reboot the device and select „Auto Control“.
Select Options, select „Load File Slot 0“ to load „WT32SC01_ESPNOW_DRV8833_V1_0“.
Save the loaded record to another file slot of your choice (1 - 19).

About Steering:

There is no real steering with the used rc-car toy. Steering is implemented in three modes.

1. Forward, turn hard right - Drive the left motor forward and stop brake the right motor.
2. Forward, turn soft right - Drive the left motor forward and stop coast the right motor.
3. Forward, turn in place right - Drive the left motor forward and the right motor reverse.

1. Forward, turn hard left - Drive the right motor forward and stop brake the left motor.
2. Forward, turn soft left - Drive the right motor forward and stop coast the left motor.
3. Forward, turn in place left - Drive the right motor forward and the left motor reverse.

The similar procedure if driving reverse. Playing around with PWM could be an option.

For the third option powerful motors / gears are necessary, and also a good power supply. With the here used rc-car toy the third option isn‘t possible, all components together are to heavy, the motors are not strong enough, the tires are not smooth enough...
Removing all tires the third option works on a smooth floor, but that crazy modification is nothing for the real life.
An alternative could be a RC-Car toy with mecanum wheels (~ 25 Euro at Amazon).

About Recording:

Recording was implemented on the last meter, just for the idea. Take it as a gimmick.
Don't expect high precision recording, there are to much environmental influences.
The tires, the motors, the gears, the touchscreen behavior, the battery power, the floor...

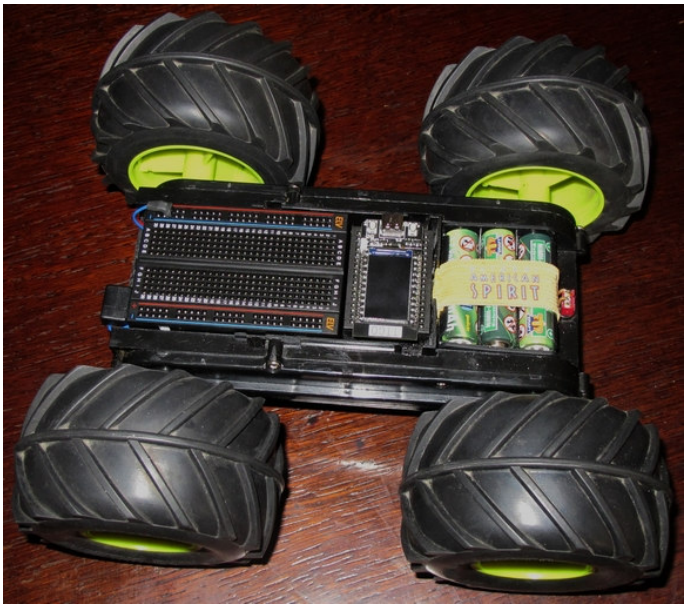
ESP32 Robot 1 - Arduino Software

ESP32-NOW, LILYGO TTGO-T-Display (Receiver) – V1

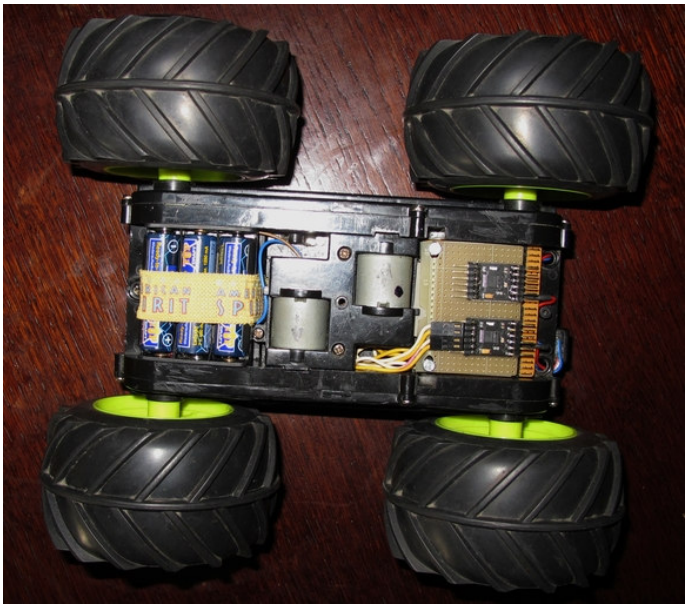
Documentations and Libraries:

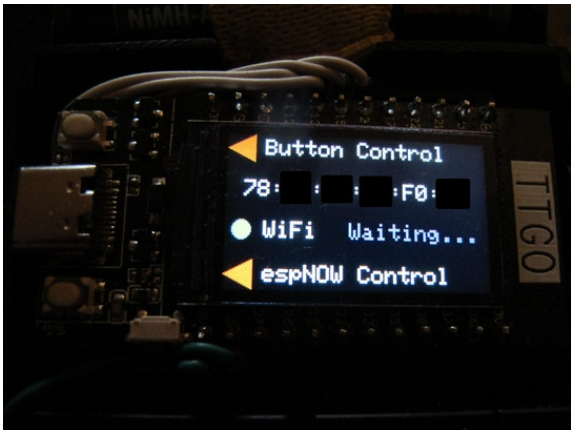
Overview ESP32 - https://socialcompare.com/en/comparison/esp8266-vs-esp32-vs-esp32-s2
Espressif Docs - https://www.espressif.com/en/support/documents/technical-documents
LILYGO TTGO-T - https://github.com/Xinyuan-LilyGO/TTGO-T-Display
TFT ST7789V - https://newhavendisplay.com/content/datasheets/ST7789V.pdf
TFT_eSPI library - https://github.com/Bodmer/TFT_eSPI
ST7789V library - included in the TFT_eSPI library

Robot – Top



Robot – Bottom





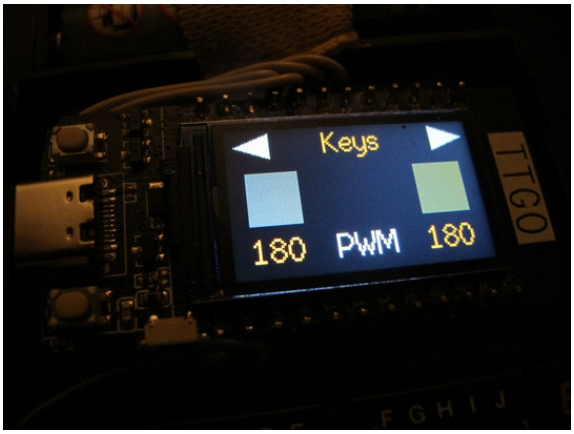
Main Menu:

If Button Control selected: WiFi is disabled

If ESP-NOW Control selected: WiFi is enabled

The displayed MAC address must be inserted in the WT32-SC01 transmitter software, before compiling.

The software is waiting for incoming ESP-NOW data. Received ESP-NOW data will be automaticly executed. It is not necessary to select ESP-NOW Control.



Button Control – Show-Screen:

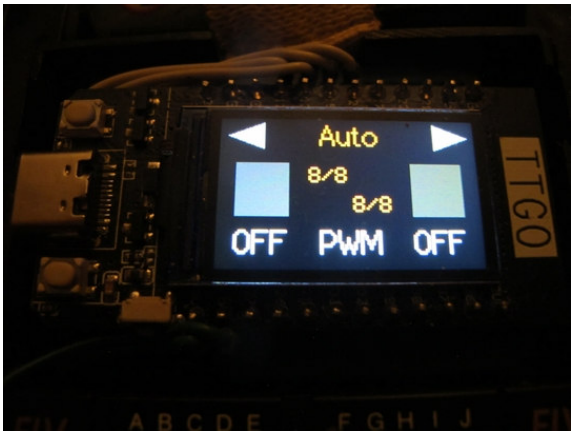
Displays the mode, the current step, the motor directions and the PWM, in real time.

Short press upper button changes left motor PWM

Long press upper button changes left motor direction

Short press lower button changes right motor PWM

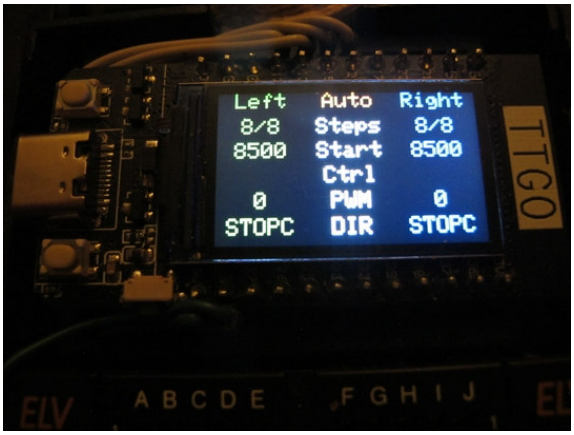
Long press lower button changes right motor direction



ESP-NOW Control – Show-Screen:
Displays the mode, the current step, the motor directions and the PWM, in real time for each step.

Press the upper button for the Show-Screen

Press the lower button for the Data-Screen

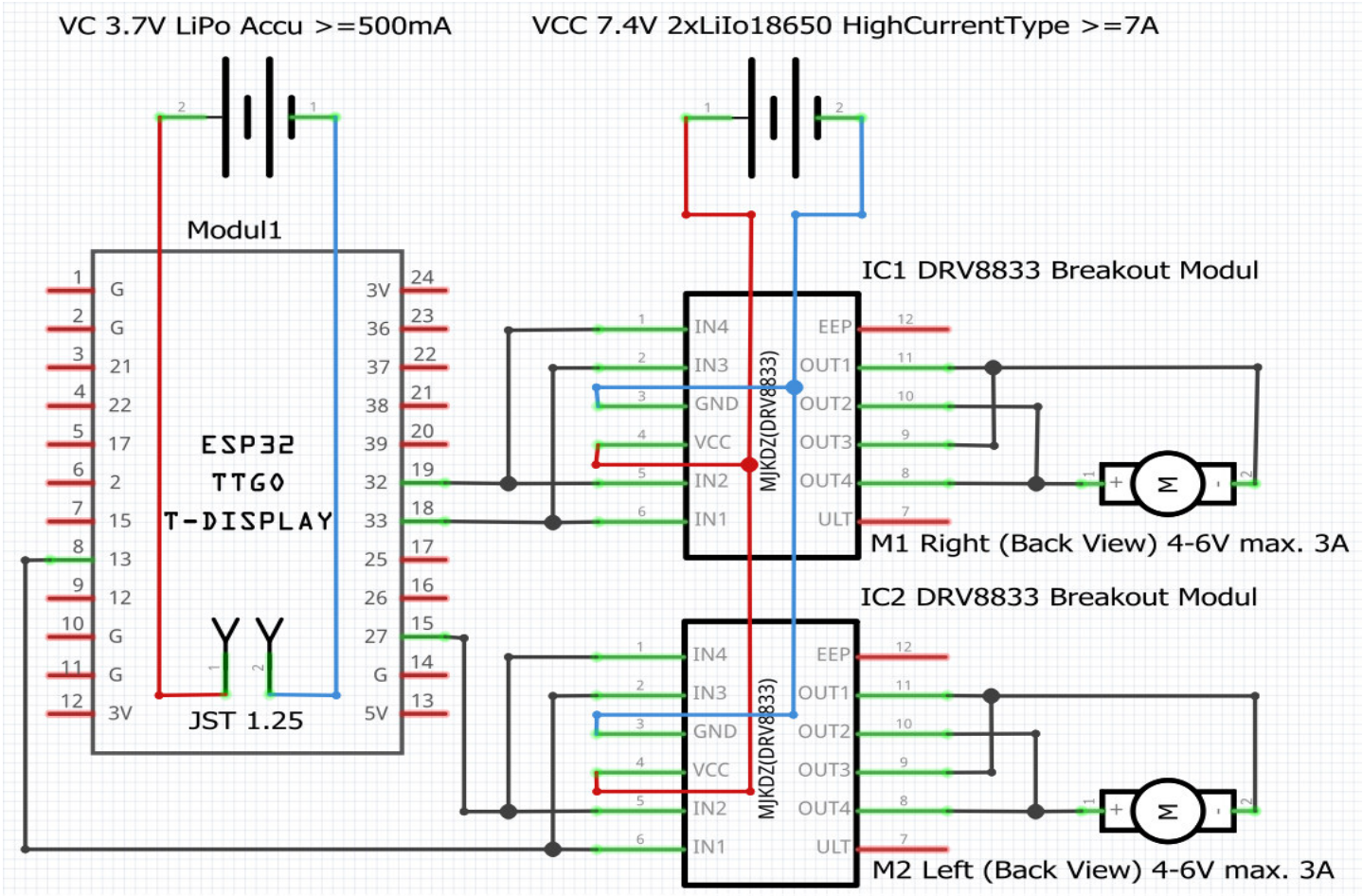


ESP-NOW Control – Data-Screen:
Displays the left and the right motor data for the step currently executed, in real time. The data screen is available for all modes, except Button Control.

Press the upper button for the Show-Screen

Press the lower button for the Data-Screen

LILYGO TTGO-T-Display receiver and motor driver schematic:



Addendum:

The chassis of the robot is a 30 years old RC car toy (chassis 190mm x 93mm, wheel width 60mm, wheel diameter 105mm) with 2 DC-Motors (Startup: 1.2A - 2.2A, Free Running: 0.5A, Working: 0.9A - 2A, Blocking: up to 2.7A).
Feel free to use another or a new RC-Car toy. A RC-Car toy with mecanum wheels is a cheap and very nice option. Make sure that enough space for all components is available.

The motor driver is a DRV8833 breakout board, 1.2A - 1.5A each channel.
Two DRV8833 breakout boards (two H-bridges each) are mounted.
The two H-bridges in one DRV8833 are connected parallel for double the current of a single H-bridge. The DRV8833 works from 2.7 (should be >= 4.8V) to 10 Volt.
No extra power supply for the DRV8833 logic necessary.

A self made power board is located under the motor driver board with some pinheads for quick access, from left to right, GND, 7.2V (Accus), 5V (Powerbank) and 3.3V (LD1117A).

Notes:

The ESP-NOW communication between the WT32-SC01 Development Board (Transmitter) and the LILYGO TTGO-T-Display Development Board (Receiver) works without a problem.
If the motor selection, left / right, is wrong swap the motor wires.
If the rotation direction of one or both motors is wrong swap the IN wires of the DRV8833.

The powerbank (2300mAh, 5V max. 1A), located under the breadboard, is extra thin (7mm). Powerbank and breadboards are mounted with double-faced adhesive tape.
Anyway, the famous Lilygo LILYGO TTGO-T-Display device offers a battery connector with a charging and a discharging option. It seems to be a better solution to use a Li-Po 3.7V battery.

Using six mignon Ni-Mh accus and batteryholders for the power supply of the motors is not a good solution, but only a standard charger is required. If possible use a soldered accu-pack, but a special accu-pack charger is required. Two Li-Io accus (18650) are the best solution, use a high current typ (>= 7A).

Do not use the L293 / L298 motor driver. That chips are very, very old and the voltage drop can reach 3V. The voltage drop of the DRV8833 is about 0.3V.

In this example the DRV8833 current limitation option isn't implemented. Read the DRV8833 datasheet. In that case the DRV8833 breakout board must be modified, which is a SMD-Task.

In the worst case scenario the electrical current of a single motor, in this example, can reach more than 2.7A. For that reason all motor power supply wires and all motor wires should be >= 0,25mm², better >= 0,34mm². Finaly, if everything works as expected, all wires for the motors should be fixed soldered. The electrical contact resistance should not underestimate!

The total weight of all components must match the power of the batteries and the motors. If not the machine won‘t move as desired!

Licence: