Treadmill sideshift documentation

Version 1.0.0

Martin Kriz

August 27, 2025

Contents

List of Code Listings

1 Introduction

A PCB developed for a treadmill in the lab to provide lateral movement for testing the camera box. Two identical PCBs are used, controlling both ends of the treadmill.

2 Hardware

2.1 Block diagram

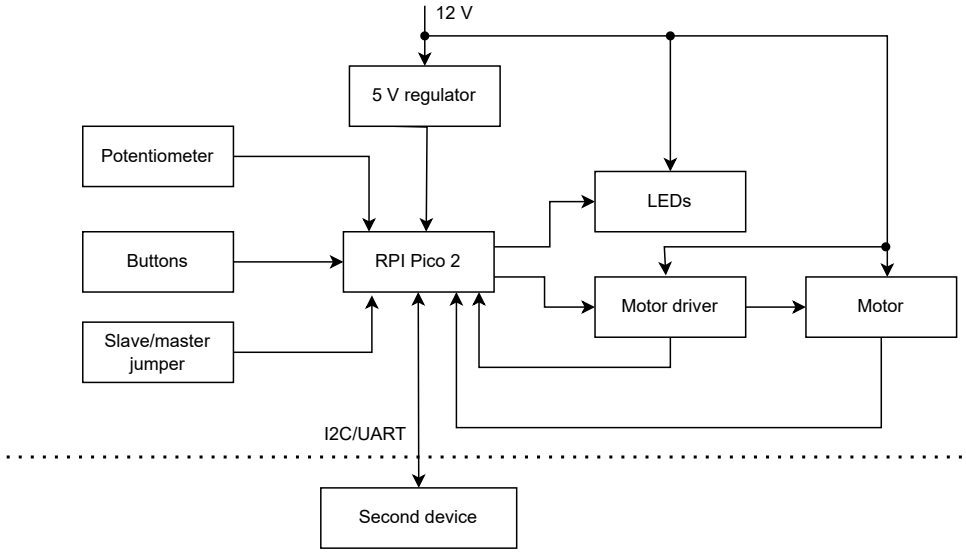


Figure 1: Block diagram of the treadmill sideshift module.

2.2 Motor Linak LA14

Exact model number is Linak 14020130000A0C06:

- **Feedback:** Hall potentiometer
- **Motor type:** 12 V BDC Fast
- **Endstop:** Power switch

The figure 2 shows the correct connection of the motor to the PCB.

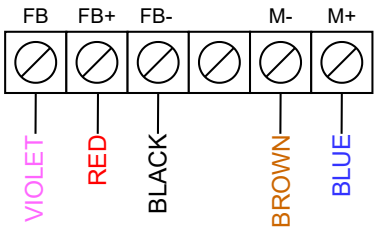


Figure 2: Connecting motor to the PCB.

2.3 Motor driver DRV8873

Configuration of the DRV8873 used in this project:

- **Interface:** Hardware
- **Current sensing:** Active over 360 Ohm resistor
- **Control mode:** PH/EN
- **PWM:** Software use 25 kHz
- **Slew rate:** $13V/\mu s$
- **Current ITRIP regulation:** Enabled
- **Open load diagnostic:** Enabled

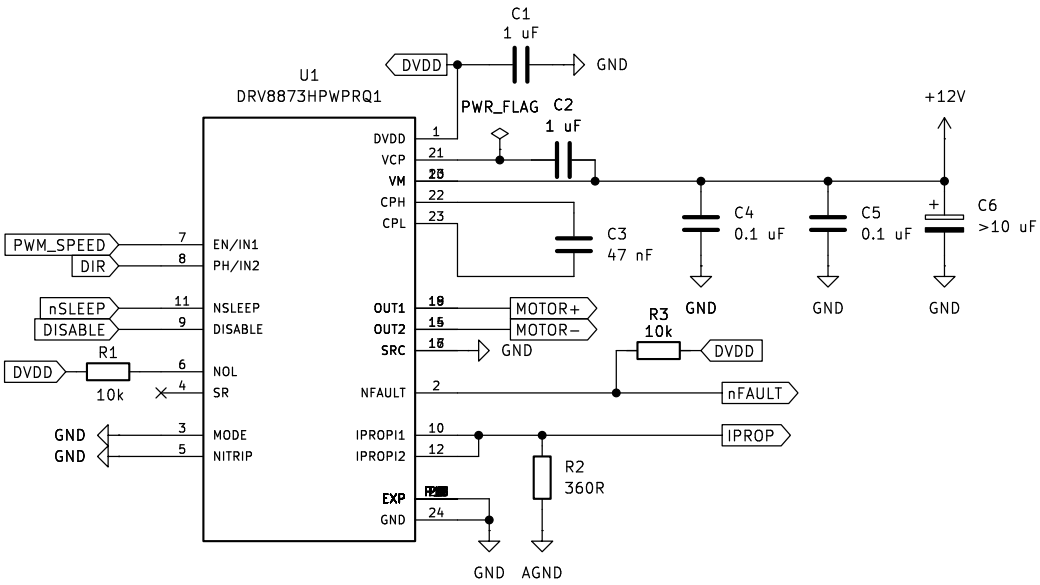


Figure 3: DRV8873H schematic.

2.4 RPi Pico 2

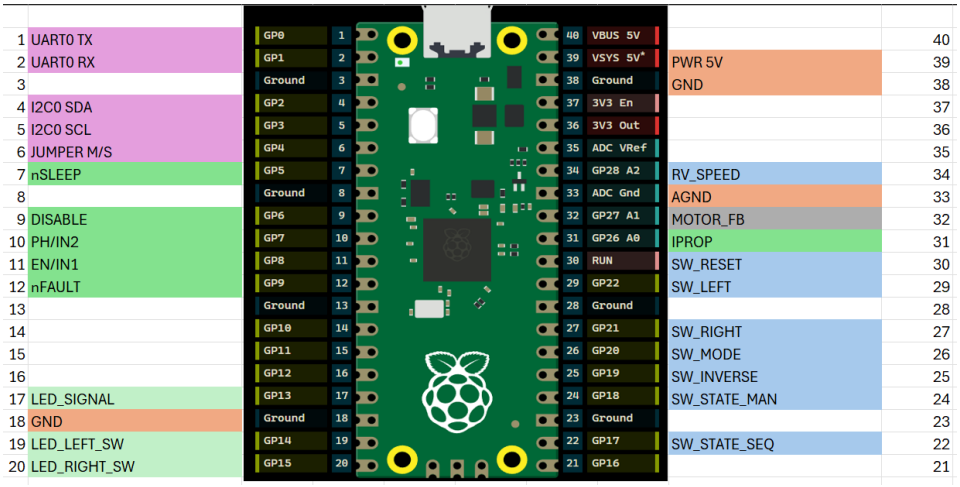


Figure 4: RPi Pico pin mapping.

The Raspberry Pi Pico is powered from a 5V regulator and connected through a Schottky diode so that when a USB cable is connected to the Pico, it cannot supply power to the rest of the PCB.

A reset button is placed on the PCB, and when pressed, it grounds the RUN

pin, causing the Pico to reset.

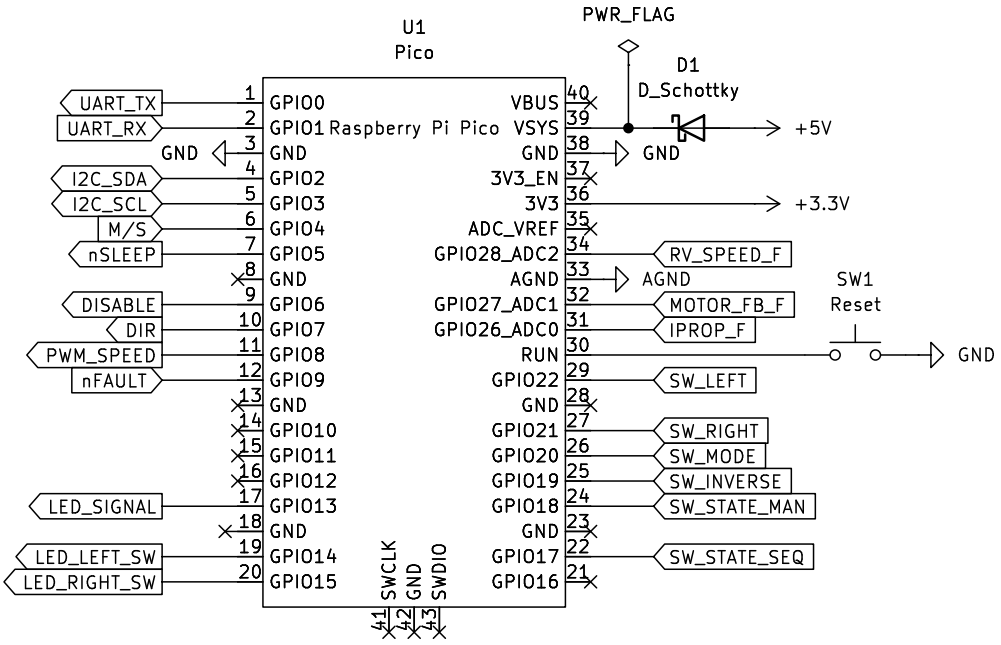


Figure 5: RPi Pico schematic

2.5 Power supply

An external 12 VDC power supply is connected to the board, which should be able to provide a current of at least 1.5 to 2 A.

The power input is 12 V is protected by a 5 A fuse based on Table 1, providing protection against reverse polarity and short circuits. In case of reverse polarity, the fuse must be replaced as it will be permanently damaged. As a 5V regulator, the LM7805 is used, and an LED is connected to it to indicate the voltage at the regulator's output. See the schematic in figure 6.

Component	Supply Voltage (V)	Max Current (mA)	Power Source
RPI Pico	5	100	Regulator (7805)
Driver DRV8833	12	10	Power supply
DC motor	12	2500	Driver
Motor feedback	12	1	Power supply
State LED	12	20	Power supply
Mode button LED	12	20	Power supply
PWR switch LED	12	20	Power supply

Table 1: Power consumption of individual components

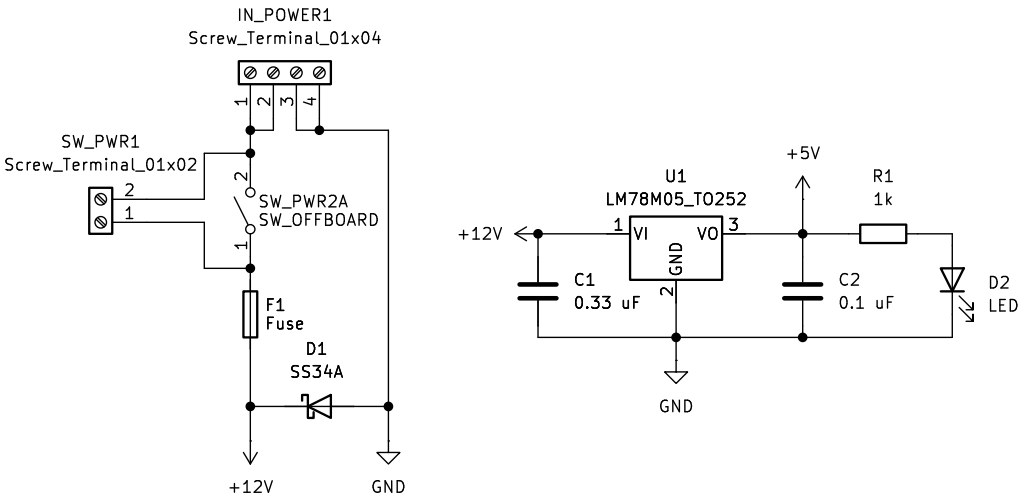


Figure 6: Power supply schematic.

2.6 Analog circuits

All analog signals use simple RC low-pass filters, and their cutoff frequencies are listed in table 2. All analog sections also have a separate ground, AGND.

Signal	R	C	Cutoff frequency
Motor feedback	10 kΩ	1 μF	15.9 Hz
Potentiometer	10 kΩ	1 μF	15.9 Hz
Motor current	10 kΩ	3.3 nF	4823 Hz

Table 2: RC Low-pass filters for analog signals

2.6.1 Motor current IPROP

The motor current is measured using the IPROPI1 and IPROPI2 pins, which are connected together for the purpose of this application. For converting the current into a voltage, the resistor R_{sense} is used, and its value is determined from the following equation:

$$R_{sense} = \frac{k \cdot U_{max}}{I_0} = \frac{1100 \cdot 3.3}{10} \doteq 360\ \Omega \tag{1}$$

2.6.2 Motor feedback

The motor feedback from the LINAK LA14 ranges from 0 to 10 V; a voltage divider is used to scale it down to 0–3.3 V for the Raspberry Pi Pico's ADC. In addition, a safety margin is provided so that in the event of a fault, where a voltage of up to 12 V may appear on the pin, the ADC will not be damaged.

$U_{in}[V]$	R_1 [Ω]	R_2 [Ω]	$U_{out}[V]$
12	100k	36k	3.18

Table 3: Voltage divider for motor feedback input.

2.7 LEDs

For panel mounting, the LEDs need to be powered from 12 V. Therefore, the ULN2004A integrated circuit was used. To set the LED current to approximately 20 mA, a series resistor of 470 Ω was chosen.The LED lights up when a logic high (1) is applied to the corresponding Raspberry Pi Pico pin.

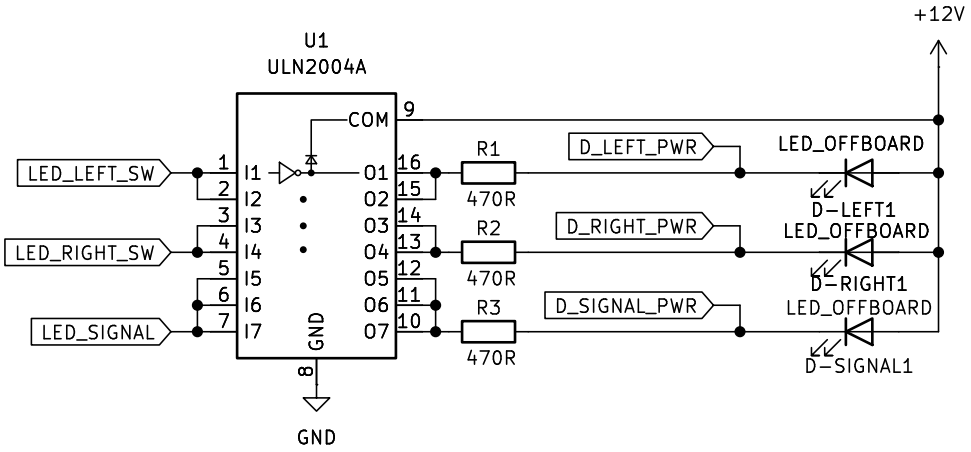


Figure 7: LEDs schematic.

2.8 Communication

Communication with the second board is carried out via I^{2}C or UART. When using the I^{2}C bus, it is necessary to set whether the device acts as a master or a slave by means of a hardware jumper. It is necessary to ensure that one board has the jumper set to the slave position and the other board has the jumper set to the master bus position.

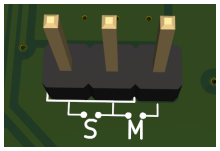


Figure 8: Master/Slave jumper.

For PCB version v1.0.0, the pin selected for I^{2}C is incorrect, and it is not possible to operate it on the prepared pins. As a temporary solution, the I^{2}C protocol can be used on the UART pins, where Tx corresponds to SDA and Rx corresponds to SCL. Therefore, when switching protocols, it is always necessary to swap the wires.

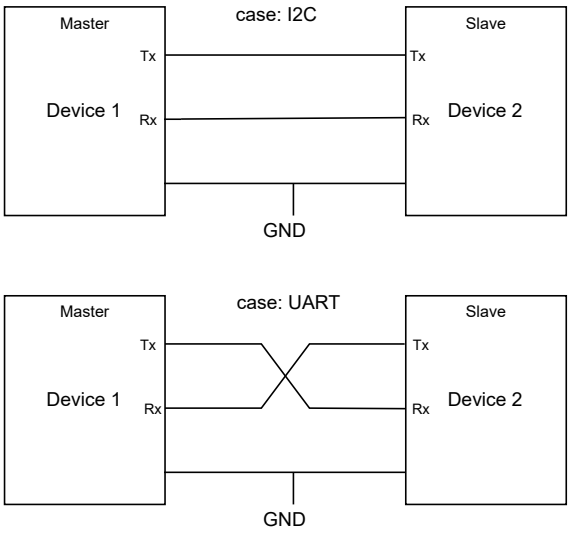


Figure 9: Temporary communication wiring.

3 Control panel

3.1 Front view

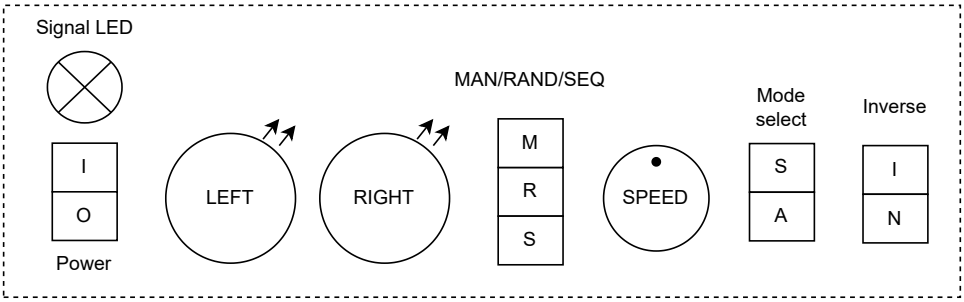


Figure 10: Control panel front view.

3.2 Components

Signal LED 12 V, 20 mA

Power switch A rocker switch rated for at least 2 A

L/R button A non-latching push button with 12 V LED backlight.

State switch A three-position switch

Speed potentiometer 10 kΩ potentiometer

Mode/Inverse switch A two-position switch

3.3 Wiring

Each screw terminal for connection is marked with the number 1 on one side; numbering proceeds from that side. An example is shown in figure 11.

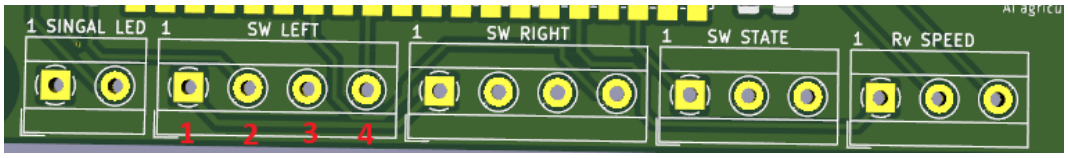


Figure 11: Screw terminal numbering.

3.4 Usage

After turning on the power using the **power button**, calibration starts automatically. Until the calibration is complete, the device cannot be controlled.

The higher a button is placed in this list, the higher its priority.

Mode switch Switches between the state of following the second device synchronous operation (SYN) or the asynchronous state (ASYN), where it is possible to control the device using the control panel.

State switch Three-position switch, allowing switching between manual control, random, and sequential modes. This switch is considered only when ASYN is selected on the previous button.

L/R buttons Buttons used to manually move the piston while held. These buttons are active only when the previous switch is set to the manual position. When the buttons are active, the LED backlight is illuminated.

Speed potentiometer In asynchronous mode, the potentiometer can be used to set the maximum speed of the piston movement.

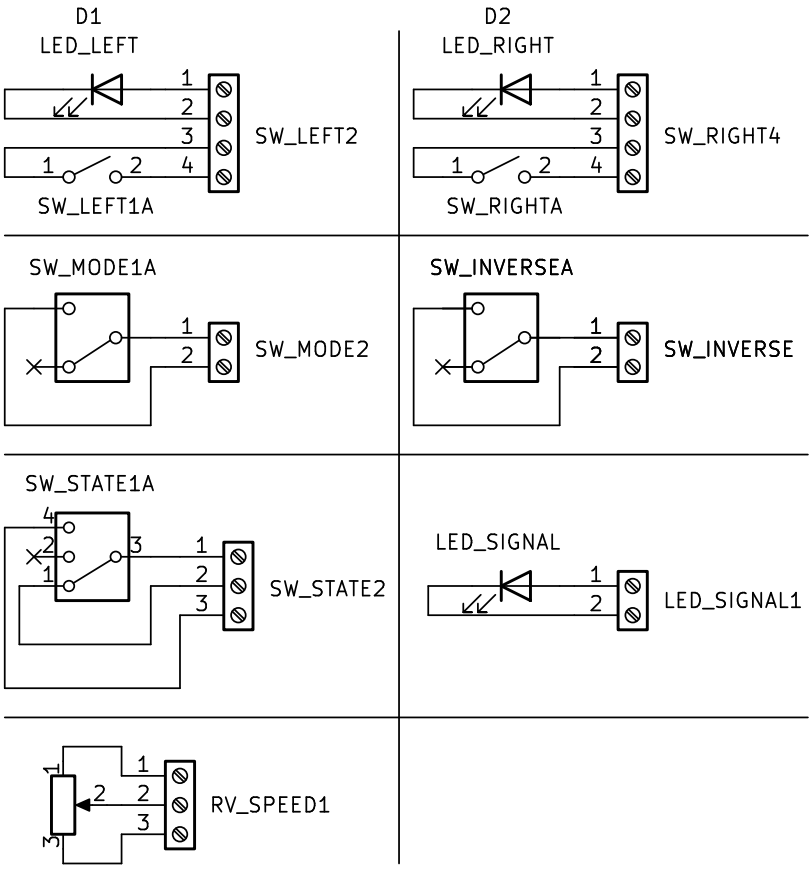


Figure 12: Control panel connection diagram. Note: This is a connection-only schematic.

The signal LED is active when the device is in any mode other than manual, indicating that the device is active. If the device is in synchronous mode and the LED is blinking, this indicates a communication error.

4 Software

The code is written in the Arduino IDE for the Raspberry Pi Pico2. The project structure is as follows:

```
treadmill_sideshift.ino ..... Main project file
src/
include/ ..... Classes used by main hardware handling, algorithms
    I2CCommunication.h .....Provide I2C communication
    ICommunication.h .....Defines communication interface
    Motor.h ..... Provide communication with the motor driver
    MotorFeedback.h .....Reading analog feedback from motor
    OtherDevice.h .....Defines communication message, store the last
    message
    Pid.h .....PID controller with integral windup protection
    PositionCalculator.h .converts raw ADC readings from a motor
    piston sensor into a physical position
    UARTCommunication.h ..... Provide I2C communication
    UserInterface.h .....handles buttons, potentiometer, and LED
    indicators
test/ .....Codes for solo hardware testing, other device simulation
    ICommunication_test/
    Motor_test/
    MotorFeedback_test/
    SynMode_test/
    UserInterface_test/
```

The properties of the classes are defined in the headers of the individual files.

4.1 State machine

State description:

- **CALIB_LEFT** – Initial state, used to find the maximum ADC value of the transducer, which is then stored to determine the exact position of the piston at its extreme end. The maximum is determined by running the motor until its position no longer changes for a certain period of time.
- **CALIB_RIGHT** – The same, but in this case it concerns the minimum ADC value of the transducer.

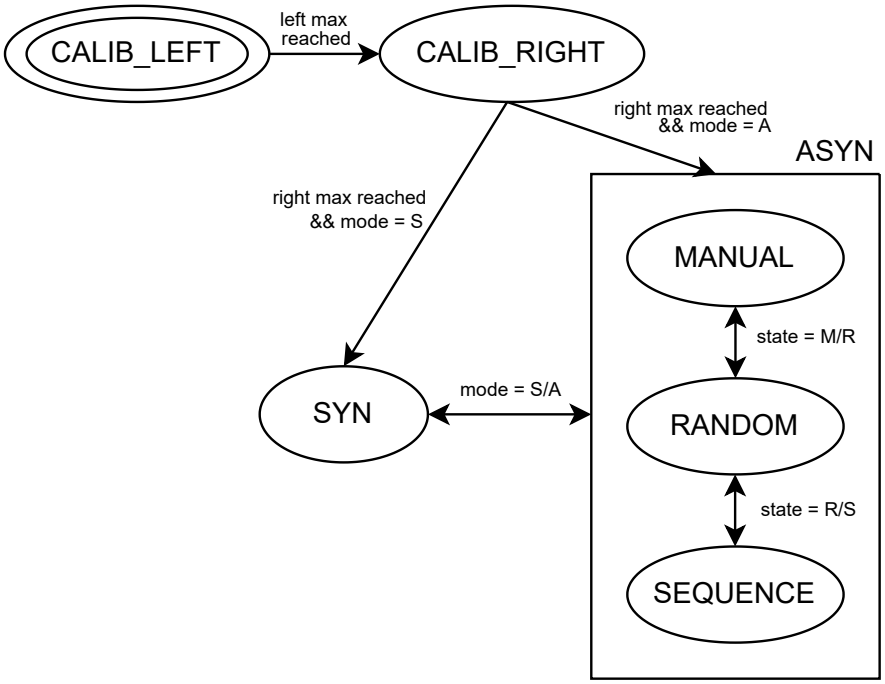


Figure 13: Sideshift state machine.

- **SYN** – State in which the piston follows the second piston (position is known via communication). A PI controller is used to maintain zero control deviation.
- **MANUAL** – Manual control is active, allowing the piston to be positioned using the L/R buttons. The maximum speed is selected via a potentiometer.
- **RANDOM** – In random mode, a position is selected within the range of the piston rod, and a PI controller drives the rod to the selected position. After reaching the position, the next position is selected randomly. Maximum speed is set via a potentiometer.
- **SEQUENCE** – The piston rod moves left and right, with the maximum speed set via a potentiometer.

Transitions between states

- **MAX REACHED** – The maximum (minimum) is considered reached if the motor is driven in one direction, and the ADC converter value does not change by more than the interval defined in the constant `changeTolerance` during the time interval defined in the constant `stableThreshold`.
- **STATE and MODE** – These transitions depend on the position of the switches on the control panel.

4.2 Communication

4.2.1 Choosing protocol

For communication, it is necessary to correctly select the protocol, which is chosen in the `setup` loop. One of the following lines must be uncommented:

```
comm = new UARTCommunication();
//comm = new I2CCommunication(isMaster);
```

By enabling the UART protocol, a class is created that handles the communication. The protocols must be the same on both boards.

4.2.2 Message type

Messages are sent as strings. The `OtherDevice` class handles this conversion, and follows these rules:

- The message always starts with the character `<`.
- Values are separated by commas.
- The first value is the piston position (`pos`), represented as a floating-point number with two decimal places.
- The second value is the potentiometer reading (`pot`), represented as an integer.
- The third value is the direction (`dir`), represented as an integer.
- The message always ends with the character `;`.

An example of a generated message:

```
<12.45,75,1;
```

Meaning of the message:

- `pos = 12.45` → piston position in millimeters.
- `pot = 75` → potentiometer value in percent (75 % of maximum speed).
- `dir = 1` → piston movement direction.

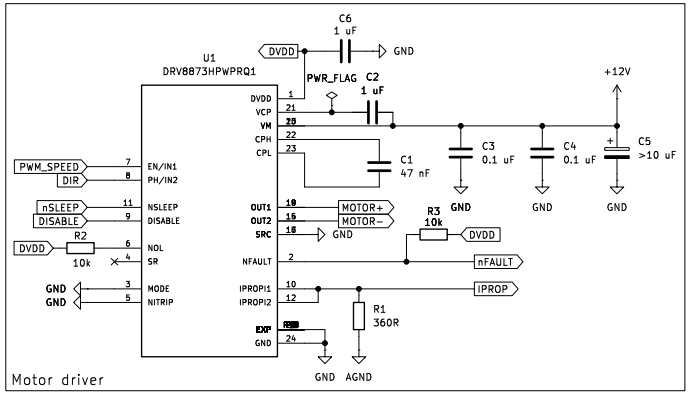
4.2.3 Communication period

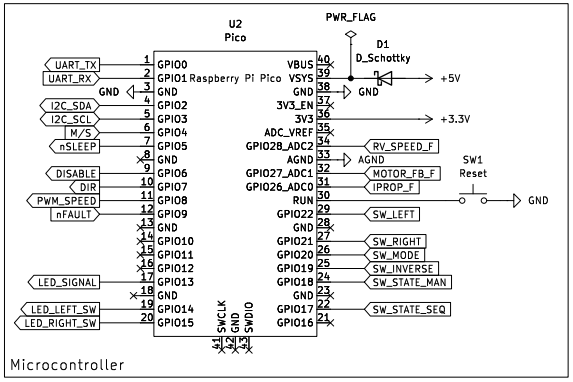
The communication period can be set by the constant *COMM_MS* in the main file. By default, it is set and tested to 50 ms.

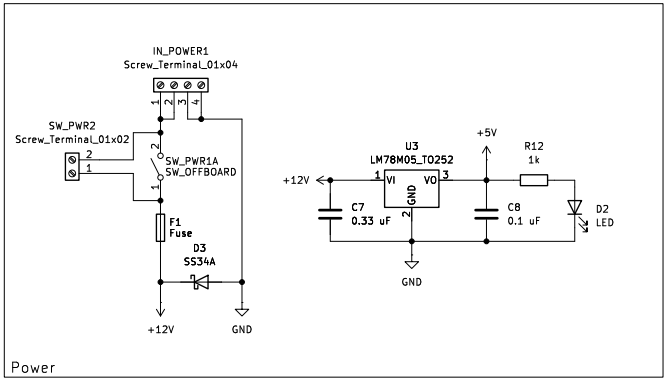
5 Appendix

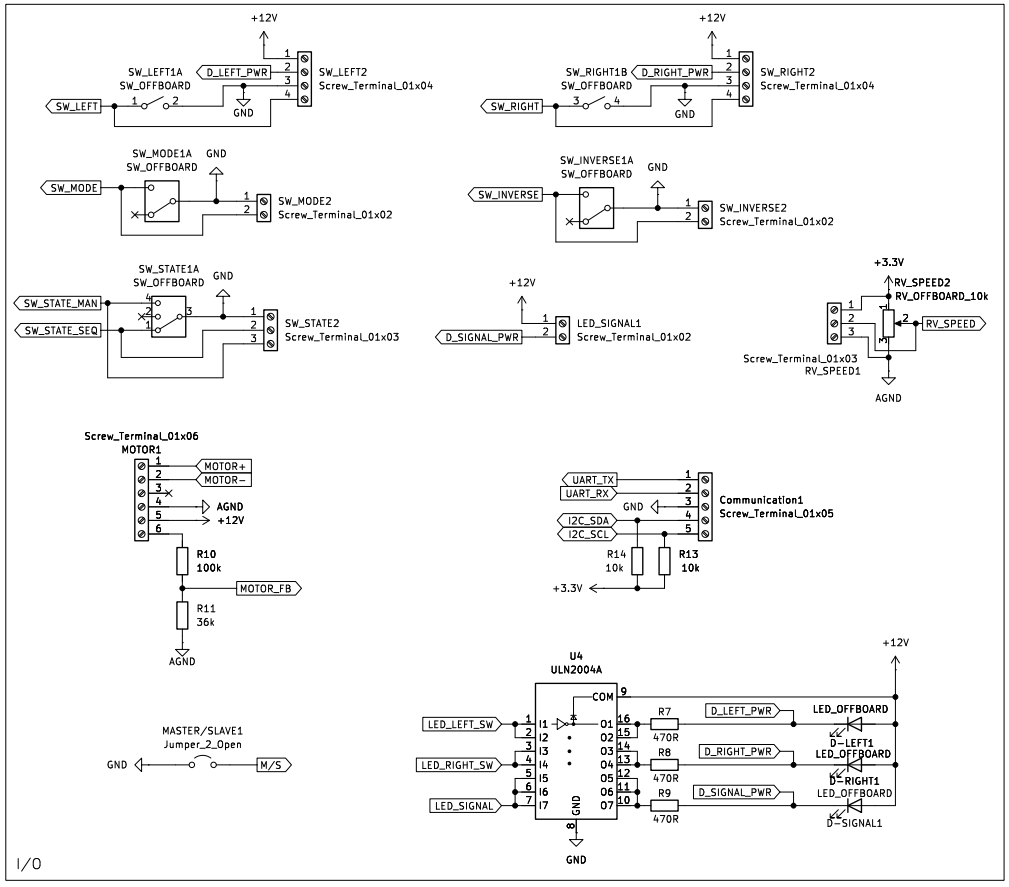
A Schematic diagram

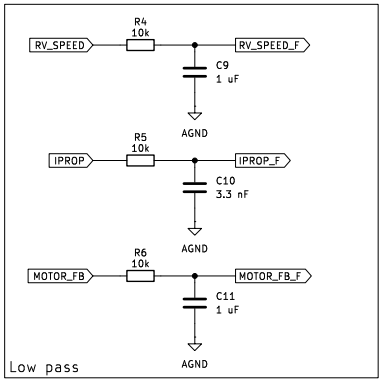
B PCB layout TOP (red) and BOTTOM (blue)

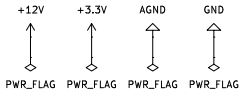

Motor driver
U1
DRV8873HPWPRQ1
C6
1 uF
C2
1 uF
C1
47 nF
C3
0.1 uF
C4
0.1 uF
C5
>10 uF
R1
360R
R2
10k
R3
10k
DVDD
GND
+12V
PWR_FLAG
PWM_SPEED
DIR
nSLEEP
DISABLE
MOTOR+
MOTOR−
nFAULT
IPROPI
AGND

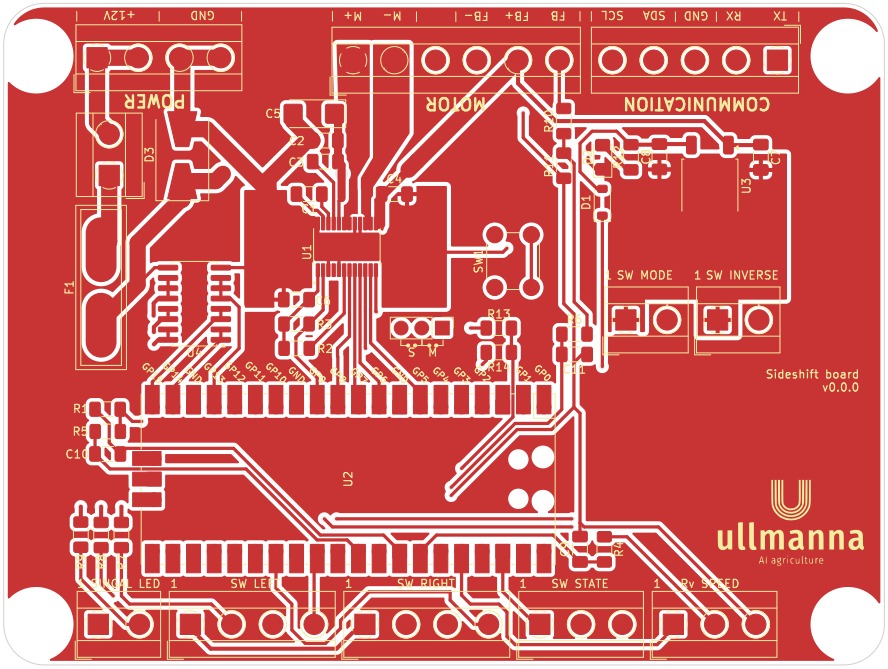

Microcontroller
U2
Pico
Raspberry Pi Pico
PWR_FLAG
D1
D_Schottky
+5V
+3.3V
SW1
Reset
UART_TX
UART_RX
I2C_SDA
I2C_SCL
M/S
nSLEEP
DISABLE
DIR
PWM_SPEED
nFAULT
LED_SIGNAL
LED_LEFT_SW
LED_RIGHT_SW
RV_SPEED_F
MOTOR_FB_F
IPROP_F
SW_LEFT
SW_RIGHT
SW_MODE
SW_INVERSE
SW_STATE_MAN
SW_STATE_SEQ
SWCLK
SWDIO

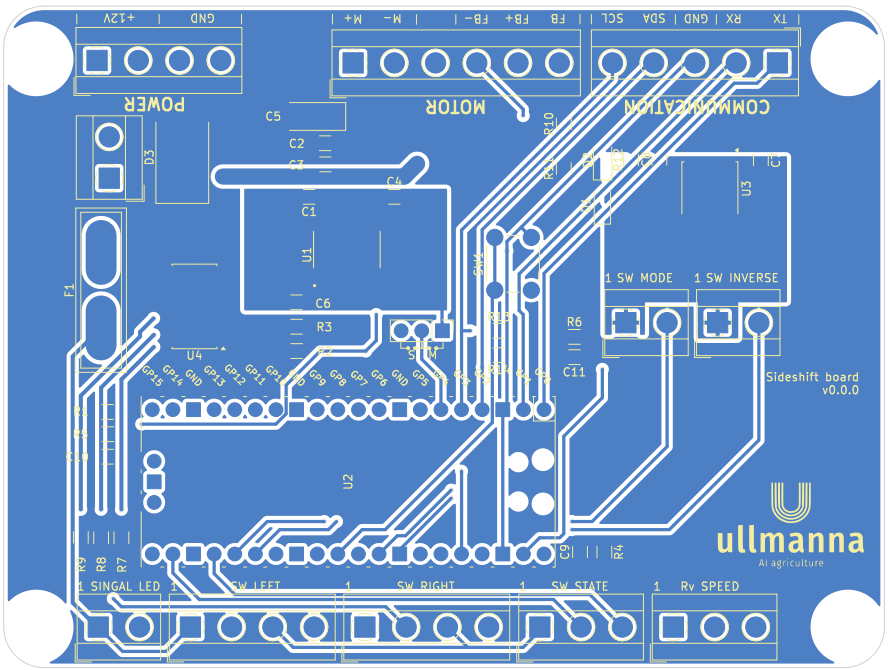

Power
IN_POWER1
Screw_Terminal_01x04
SW_PWR2
Screw_Terminal_01x02
SW_PWR1A
SW_OFFBOARD
F1
Fuse
D3
SS34A
U3
LM78M05_TO252
C7
0.33 uF
C8
0.1 uF
R12
1k
D2
LED
+12V
+5V
GND


I/O
SW_LEFT1A
SW_OFFBOARD
D_LEFT_PWR
SW_LEFT2
Screw_Terminal_01x04
SW_RIGHT1B
SW_RIGHT2
SW_MODE1A
SW_MODE2
Screw_Terminal_01x02
SW_INVERSE1A
SW_INVERSE2
SW_STATE1A
SW_STATE2
Screw_Terminal_01x03
LED_SIGNAL1
RV_SPEED2
RV_OFFBOARD_10k
RV_SPEED1
Screw_Terminal_01x06
MOTOR1
R10
100k
R11
36k
Communication1
Screw_Terminal_01x05
R14
10k
R13
10k
MASTER/SLAVE1
Jumper_2_Open
U4
ULN2004A
COM
R7
R8
R9
470R
D_LEFT1
D_RIGHT1
D_SIGNAL1
LED_OFFBOARD
+12V
+3.3V
AGND
GND


Low pass
R4
10k
C9
1 uF
R5
10k
C10
3.3 nF
R6
10k
C11
1 uF
RV_SPEED
RV_SPEED_F
IPROP
IPROP_F
MOTOR_FB
MOTOR_FB_F
AGND


+12V
+3.3V
AGND
GND
PWR_FLAG


+12V GND M+ M- FB- FB+ FB SCL SDA GND RX TX
POWER
MOTOR
COMMUNICATION
1 SW MODE
1 SW INVERSE
Sideshift board
v0.0.0
ullmanna
AI agriculture
1 SINGAL LED 1 SW LEFT 1 SW RIGHT 1 SW STATE 1 Rv SPEED


+12V GND M+ M- FB- FB+ FB SCL SDA GND RX TX
POWER
MOTOR
COMMUNICATION
1 SW MODE
1 SW INVERSE
Sideshift board
v0.0.0
ullmanna
AI agriculture
1 SINGAL LED 1 SW LEFT 1 SW RIGHT 1 SW STATE 1 Rv SPEED