# Ultrasonic Anemometer Reference Manual

Features

* Measurement of airspeeds up to 50 m/s
  + Between 50 m/s and 60 m/s the sensor returns 50 m/s.  
    After 60 m/s the sensor returns nonsensical data!
* Analog or digital output
* Triggered or automatic measurement mode
* Averaged or double shot measurement
* Sub measurement frequency of up to 250 Hz
* Maximum average time 1s
* USB-Serial interface
* Low power consumption 16 -24 V, 100 mA

A black and white drawing of a circuit board

Description automatically generated

**PCB overview:**



**PCB first steps:**

* Jumper besides SWD connector set to the arrow position
* All other jumpers must be set to the position nearest the board edge
* Solder the analog ground jumper on the other PCB side
* **Check the resistances before applying power**
  + GND –> -12 V
  + GND –> -5 V
  + GND –> 3.3 V
  + GND –> 5 V
  + GND –> V\_drive
  + GND –> 12 V
  + Transceivers pad terminals towards each other ~ 23 Ohm
  + A0 -> VCC
  + A0 -> A1 ~ 33 kOhm
  + A1 -> A2 ~ 33 kOhm
  + A2 -> A3 ~ 33 kOhm
  + A3 -> A4 ~ 33 kOhm
* **Connect to lab bench power supply at 24V and current limit of 100mA**
  + Current should be lower than 40mA in idle
  + Power LED should be on, if the power LED is off and current near zero the PTC fuse has been triggered
* **Check all voltages:**
  + -12 V, 12 V
  + -5 V, 5 V
  + 3.3 V
  + V\_drive ~ 8V

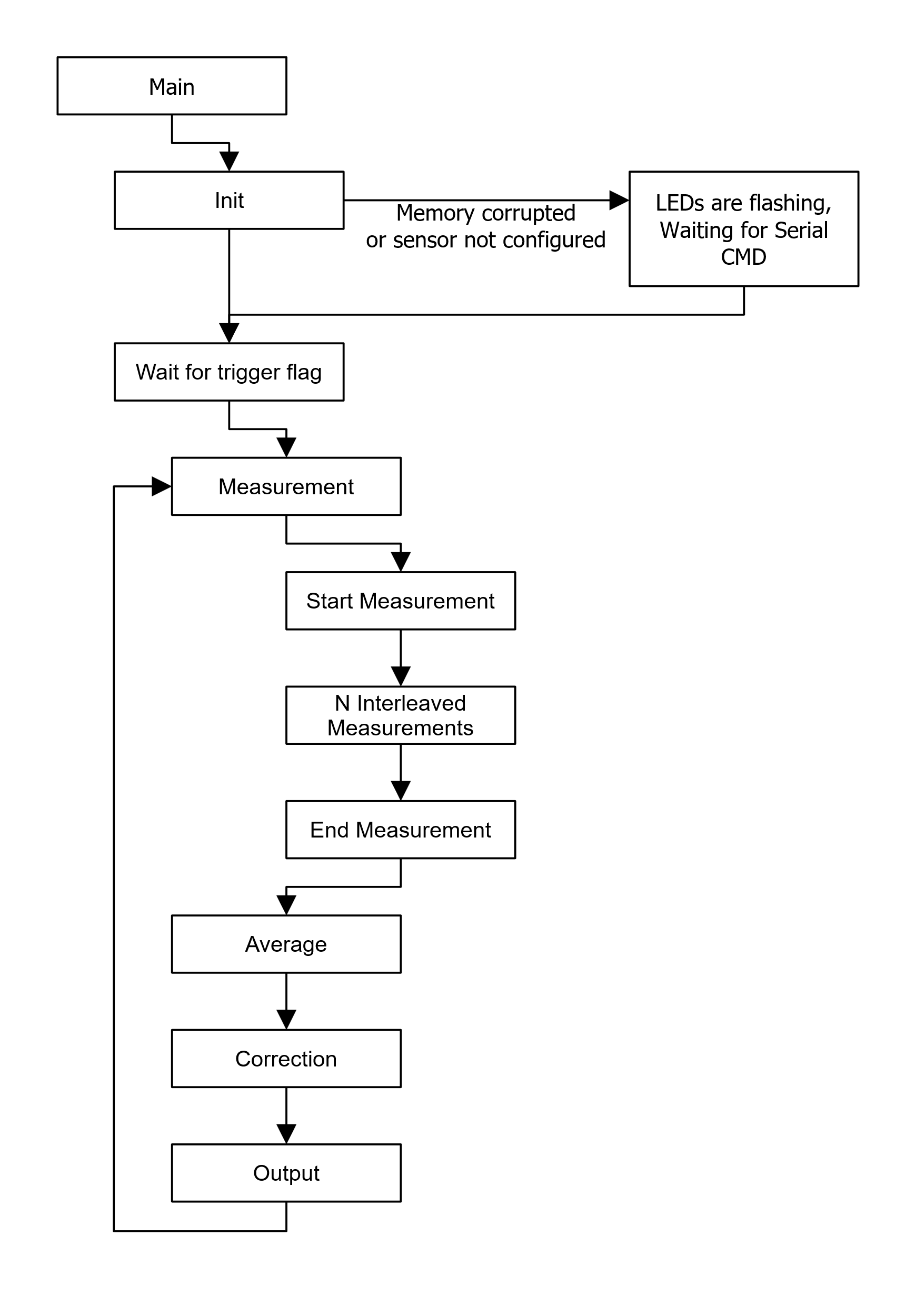
**The sensors serial interface** has a baud rate of 1Mbaud and the following commands:

|  |  |  |
| --- | --- | --- |
| Command (Name) | Options [Var\_type (Var\_name, unit), …] | Return |
| -h (Help) | - | List of commands |
| -t (Trigger) | [ f (f\_trigger, Hz), f (d\_trigger, %)],  **f\_trigger** -> Max. external trigger frequency: f\_trigger = 0 -> trigger deactivated, f\_trigger < 100 Hz,  **d\_trigger** -> Duty cycle of the measurement, d\_trigger = 1 -> double shot mode, d\_trigger <= 100% | Actual measurement parameters |
| -s (Status) | - | Returns all current settings / parameters |
| -e (Exit) | - | Confirmation |
| -w (Write to EEPROM) | - | Saves new config to EEPROM |
| -n (Zero measurement) | [f (temp, °C)], must be **taken under no wind speed condition** aka in a big box or so. | Returns new t\_0 and ref speed of sound |
| -p (Ping) |  | Returns Sensor name  and puts out a +9V, 0V, -9V signal on both DAC channels |
| -c (Calibration mode) | [i (mode)],  **mode = 0** -> first time enter calibration mode after that return raw measurement  mode = 1 -> Get calibration from PC  (mode = 2 -> read calibration)  mode = 4 -> return raw and corrected value needs a trigger [f32 (raw\_x), f32 (raw\_y), f32 (corr\_x), f32 (corr\_y), u32 (crc32)] in binary  mode = 11 -> set to normal mode with trigger  mode = 12 -> set to free run mode | Returns corrected / uncorrected measurement data |
| -I (Info) | [c (operation), i (ID)],  **operation = ?** -> get ID of the sensor returns 0xFFFF (65535) if unset  operation = ! -> writes **ID** to sensor, needs confirmation from user, the ID can only be written once ! | Returns ID of the sensor |
| -a (ASCII mode) | [i (mode)],  **mode = 0** -> leave ASCII output mode  mode = 1 -> enter continuous ASCII output  mode = 2 -> one time ASCII output | Only returns error or missing value massage |

If an external Trigger is applied the USAN has the following **LED error codes**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Description | LED 0 | LED 1 | DAC | Action |
| Normal behaviour | Flashing with half the trigger frequency | OFF | Voltage between -9.5 V and 9.5 V | All good |
| EEPROM error or other critical error | Flashing | Flashing | Voltage stepps | Connect to serial interface and restart sensor.  Serial interface output will give further instructions |
| Trigger to early warning | - | ON | - | Trigger was faster than expected, reconfigured measurement with higher trigger frequency can be ignored if it jumps off again |
| Measurement error | - | ON | - | Sensor has experienced an error while measuring, can be ignored if it jumps off again |
| Trigger lost warning | Constant | OFF | If in trigger mode both channels Zero other wise valid output | If |

The schematic of the software execution is as follows:



**How to set up a sensor:**

Steps:

1. Flash the sensor
2. Self-test the sensor
3. Record the zero measurement
4. Setup the sensor
5. Flash the sensor
   1. Download Segger Embedded Studio for Arm and RISC-V (<https://www.segger.com/downloads/embedded-studio/#embeddedstudio>)
      1. Version used: 8.14
   2. Downlaod STM32CubeProg (<https://www.st.com/en/development-tools/stm32cubeprog.html>)
      1. CONTINUE HERE
   3. Start Segger and install the following packages via “Tools” – “Package Manager”
      1. CMSIS 5 CMSIS-CORE Support Package,
      2. CMSIS 5 CMSIS-DSP Support Package,
      3. CMSIS 5 Documentation Package
      4. STMicroelectronics STM32F4xx CPU Support Package
   4. Connect the Flasher (STLINK-V3MINIE) with the board (marked SWD on the PCB) and the PC
   5. Power up the sensor (power LED on)
   6. Open the “2D-Anemometer-Firmware\_r0.emProject” with Segger via “File” – >“Open solution”
   7. Press “Play button” to flash program
      1. The sensor can be further configured with “Sensor\_config.h” file for different measurements path lengths or sub measurement frequencies
         1. Activate key testing mode, useful if the transceiver model has been changed (KEY\_TEST 0 to KEY\_TEST 1), testing should be done under the no airspeed condition and takes about 1 h, if to may keys do not pass due to over (>1000) or under (> 20) amplitude errors change to gain accordingly. The sensor logs the progress over the serial output at a baud rate of 1Mbaud. After completing change the default key to the new key (KEY\_DEFAULT) in the configuration file.
         2. Change transceiver distance (d\_0 in meters), and the ADC sample offset
         3. Change sub measurement frequency (default and max 200Hz), lowering the sub measurement frequency can help at higher airspeeds due to improved signal amplitude
      2. If needed, rebuild project
6. Zero measurement
   1. Open a serial terminal with for example the Arduino IDE, and a baud rate of 1Mbaud
   2. Must be taken under a zero windspeed condition, and known speed of sound, e.g., in a big enough box (50cmx50cmx50cm) too small boxes lead to bad filter results
   3. The -n command the zero measurement can be started
      1. If errors or warnings occur “Bad match” etc. check transceivers and transceiver voltage V\_drive
7. Calibration of the Sensor
   1. Setup:
      1. The sensor is tested on the URS75BCC turn stage following is needed:
         1. USB to RS232 Adapter
         2. SMC 100 motor controller and power supply
         3. URS75BCC and Cabel
         4. Adapter for X95 to URS75BCC, URS75BCC to Item 20x20 (URS75\_to\_Item20.stl) and sensor mounting plate (Sensor\_to\_20x20\_Item.stl)
      2. The sensor and a mini-USB cable
      3. Arduino trigger board **if** the validation should trigger an outside(external) device (like a dewetron)
      4. Velocity references
      5. Mounting:
         1. Place the URS75 (Turntable) so that the 0° is normal to the nozzle (points towards the nozzle)
         2. The sensor should be mounted with the x arrow of the coordinate system on PCB is pointing away from the zero-degree mark (in wind direction)
         3. While connecting the cables, special care should be taken to avoid tension or the cable catching on obstacles due to the sensors motion.
         4. Tape the mini-USB cable to the sensor so it can’t move, otherwise the connection to the PC is lost and the calibration has to be restarted!
   2. Start the Calibration using the MATLAB script (“Sensor\_calibration\_and\_validation.m”) and follow the instruction, calibration should take about an hour minutes
8. Setup
   1. With the -t command the trigger must be configured
      1. -t will put the sensor in a triggerless state
      2. -t 20 100 will put the sensor in mode where it expects a 20 Hz trigger and measures for 100% of the trigger period
   2. Save everything with the -w command!
   3. Press -e to exit, a small overview over the configuration should be shown

**Firmware overview:**

Main.c -> Start point of application and high-level process

Unittest.c -> Start up and Serial interface code

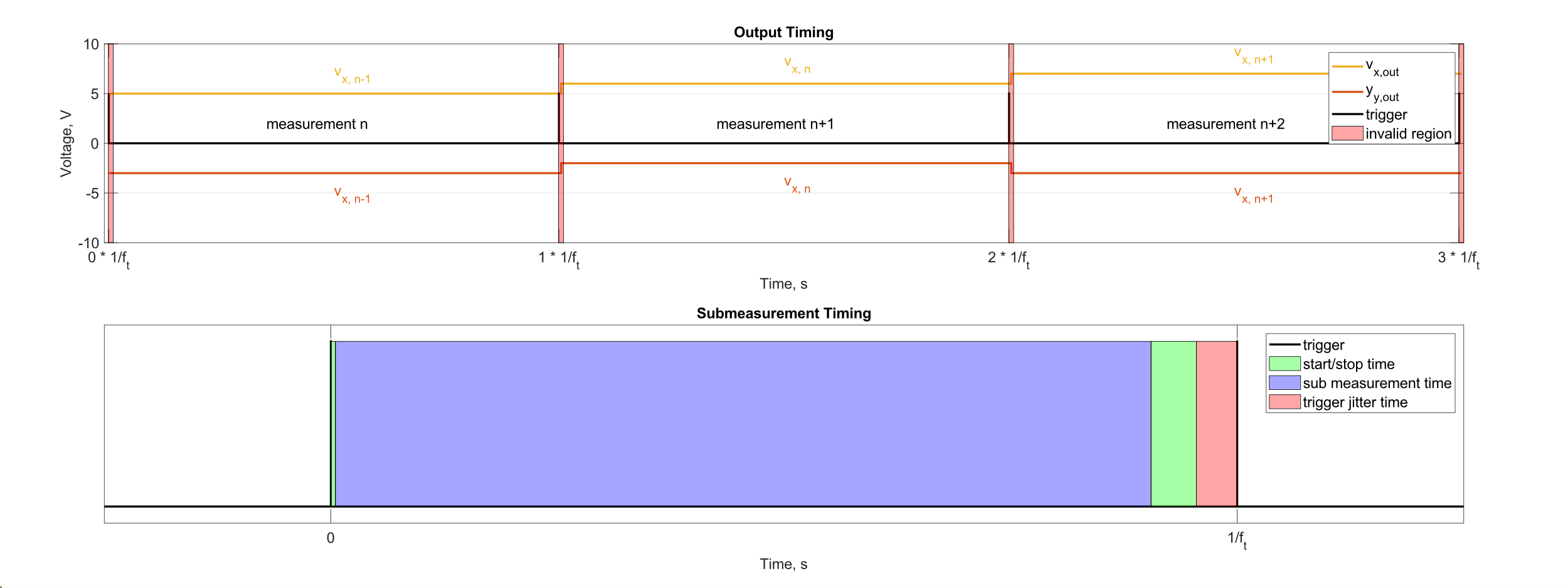
Measurement.c -> Measurement process

* + - Matched\_Filter\_r0.c -> Applies matched filter and extracts peaks
    - Filter\_Fusion\_r0.c -> Combines two matched filters into a 1D measurement
    - Fusion\_eval\_r0.c -> Combines two 1D measurements into a 2D measurement
    - Correction.c -> Applxies correction to a 2D measurement
    - Algo\_helper.c -> Helper functions for the aforementioned functions

Drivers -> Folder of low drivers

* + - DAC\_AD5732.c -> Driver and interface code for DAC
    - EEPROM.c -> Uses part of the flash sectors of the MCU to emulate EEPROM and create persistent storage -> Used sectors are deleted from memory-map!
    - Gpio.c -> Gpio drivers
    - OPT.c -> Interface to read and write to one-time programable storage (OPT)
    - Packet\_transfer.c -> CRC32 function and old packet\_transfer code
    - Shared.c -> Collection of Shared functions and miniscule code
    - Shot.c -> Used to start recordings and unlock the data, if the recording is finished, as well as to generate the gpio patterns for transmission
    - SRAM\_IS66.c -> Driver for interfacing with the SPI RAM
    - Uart.c -> non-blocking Serial (UART) driver

**Sensor data output:**



Output timing:

If the trigger is enabled the output of the sensor is updated on rising edge of trigger signal with the value of the previous measurement. If the trigger is disabled, the output is updated as soon as the measurements is finished. Due to the time needed for the microcontroller to update the DAC and the DAC to reach a steady output state, a non-valid output region does exist. This region can be negated by using an offset to the trigger signal of around 3 ms before and after the trigger but this offset may vary depending on the method used to extract the rising edge form the measurement data.

Sub measurement timing and filtering:

Each sensor measurement consists of a minimum of 2 sub measurements and maximum of 200 sub measurements.

When the trigger is configured, the sensor calculates the time in between trigger events. To compensate for trigger jitter and clock mismatch, a margin is subtracted from this time (trigger jitter time). Furthermore, the time needed to start and stop the measurement is subtracted (start/stop time). The remaining time multiplied with the configured duty cycle is the time available for sub measurements (sub measurement time). The amount of sub measurements is then calculated as the highest amount of sub samples which can be taken (with the sub measurement frequency) in the sub measurement time.

After all measurements are taken, an error and outlier detection is performed, subsequently the remaining measurements are averaged. The outlier detection is performed by filtering for velocity changes over 25 m/s.

In a final step the measurement is corrected with a lookup matrix.

Sensor startup:

When the Sensors powers on a voltage stair is displayed on its analog outputs