P2 code for micro controller

Generated by Doxygen 1.10.0

Chapter 1

File Index

1.1 File List

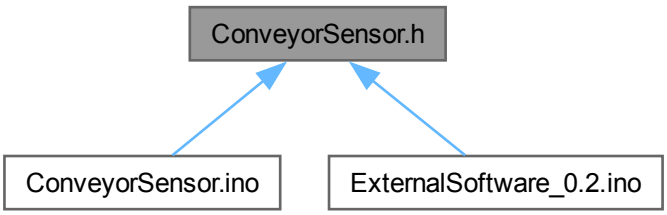
Here is a list of all files with brief descriptions:

Chapter 2

File Documentation

2.1 ConveyorSensor.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void setup_sensor ()
- void setup_motor_driver ()
- uint16_t set_normal_range ()
- bool check_if_test_is_in_front (uint16_t)
- void set_serial ()
- void run_convayor ()
- void drive_to_test ()

Variables

- const uint8_t convayor_motor_plus = 6
- const uint8_t convayor_motor_minus = 7

2.1.1 Function Documentation

2.1.1.1 check_if_test_is_in_front()

```
bool check_if_test_is_in_front (
            uint16_t normal_range )
```

Remarks

: Checs if a test is in front

Returns

: bool

Definition at line 61 of file ConveyorSensor.ino.

References sensor.

2.1.1.2 drive_to_test()

```
void drive_to_test ( )
```

Remarks

: Driving the conveyor to a new test

Returns

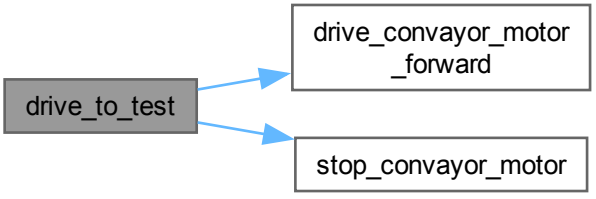
: void

Definition at line 112 of file ConveyorSensor.ino.

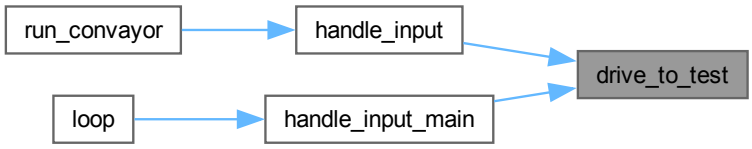
References drive_convayor_motor_forward(), sensor, and stop_convayor_motor().

Referenced by handle_input(), and handle_input_main().

Here is the call graph for this function:



Here is the caller graph for this function:



2.1.1.3 run_convayor()

```
void run_convayor ( )
```

Remarks

: Runs the conveyor system

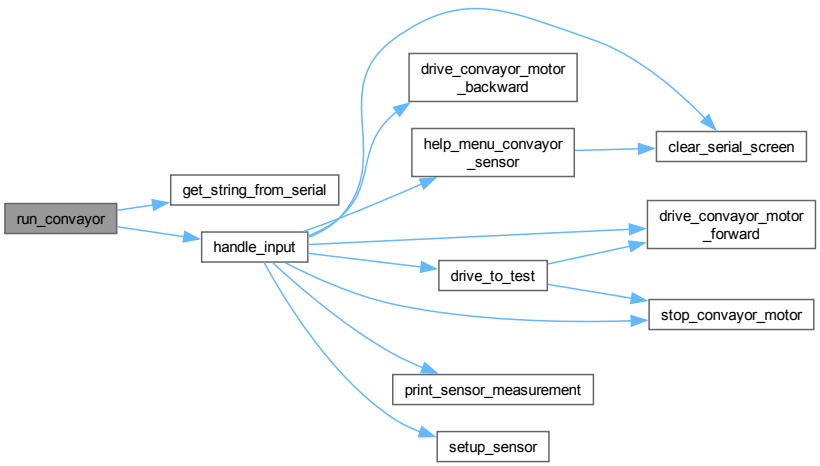
Returns

: void

Definition at line 190 of file ConveyorSensor.ino.

References get_string_from_serial(), and handle_input().

Here is the call graph for this function:



2.1.1.4 set_normal_range()

```
uint16_t set_normal_range ( )
```

Remarks

: Measureing the curent range

Returns

: millimeters

Definition at line 53 of file ConveyorSensor.ino.

References sensor.

2.1.1.5 set_serial()

```
void set_serial ( )
```

Remarks

: Setting up the Serial

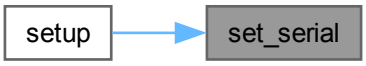
Returns

: void

Definition at line 7 of file Serial.ino.

Referenced by setup().

Here is the caller graph for this function:



2.1.1.6 setup_motor_driver()

```
void setup_motor_driver ( )
```

Remarks

: Setting up the motor for conveyor

Returns

: void

Definition at line 73 of file ConveyorSensor.ino.

References convayor_motor_minus, and convayor_motor_plus.

Referenced by setup().

Here is the caller graph for this function:

2.1.1.7 setup_sensor()

```
void setup_sensor ( )
```

Remarks

: Setting up the sensor

Returns

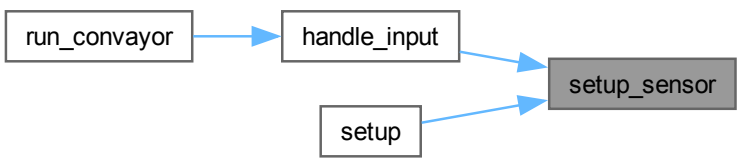
: void

Definition at line 40 of file ConveyorSensor.ino.

References sensor.

Referenced by handle_input(), and setup().

Here is the caller graph for this function:



2.1.2 Variable Documentation

2.1.2.1 convayor_motor_minus

```
const uint8_t convayor_motor_minus = 7  [extern]
```

Referenced by drive_convayor_motor_backward(), drive_convayor_motor_forward(), setup_motor_driver(), and stop_convayor_motor().

2.1.2.2 convayor_motor_plus

```
const uint8_t convayor_motor_plus = 6  [extern]
```

Referenced by drive_convayor_motor_backward(), drive_convayor_motor_forward(), setup_motor_driver(), and stop_convayor_motor().

2.2 ConveyorSensor.h

Go to the documentation of this file.

```
00001 #ifndef CONVEYORSENSOR_H
00002 #define CONVEYORSENSOR_H
00003
00004 extern const uint8_t convayor_motor_plus = 6;
00005 extern const uint8_t convayor_motor_minus = 7;
00006
00007 void setup_sensor();
00008 void setup_motor_driver();
00009 uint16_t set_normal_range();
00010 bool check_if_test_is_in_front(uint16_t);
00011 void set_serial();
00012 void run_convayor();
00013 void drive_to_test();
00014
00015 #endif
```

2.3 ConveyorSensor.ino File Reference

```
#include <VL53L0X.h>
#include "ConveyorSensor.h"
#include "Serial.h"
```

Functions

- uint16_t set_timeout_sensor ()
- uint16_t set_measurement_time_budget ()
- void setup_sensor ()
- uint16_t set_normal_range ()
- bool check_if_test_is_in_front (uint16_t normal_range)
- void setup_motor_driver ()
- void drive_convayor_motor_forward ()
- void drive_convayor_motor_backward ()
- void stop_convayor_motor ()
- void drive_to_test ()
- void help_menu_convayor_sensor ()
- void print_sensor_measurement ()
- void handle_input (String input)
- void run_convayor ()

Variables

- VL53L0X sensor

2.3.1 Function Documentation

2.3.1.1 check_if_test_is_in_front()

```
bool check_if_test_is_in_front (
            uint16_t normal_range )
```

Remarks

: Checs if a test is in front

Returns

: bool

Definition at line 61 of file ConveyorSensor.ino.

References sensor.

2.3.1.2 drive_convayor_motor_backward()

```
void drive_convayor_motor_backward ( )
```

Remarks

: Driving the conveyor backward

Returns

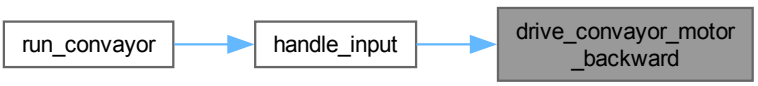
: void

Definition at line 91 of file ConveyorSensor.ino.

References convayor_motor_minus, and convayor_motor_plus.

Referenced by handle_input().

Here is the caller graph for this function:



2.3.1.3 drive_convayor_motor_forward()

```
void drive_convayor_motor_forward ( )
```

Remarks

: Driving the conveyor forward

Returns

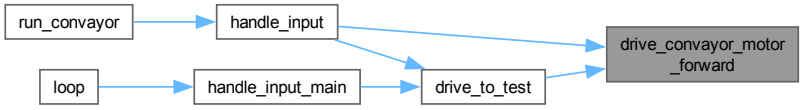
: void

Definition at line 82 of file ConveyorSensor.ino.

References convayor_motor_minus, and convayor_motor_plus.

Referenced by drive_to_test(), and handle_input().

Here is the caller graph for this function:



2.3.1.4 drive_to_test()

```
void drive_to_test ( )
```

Remarks

: Driving the conveyor to a new test

Returns

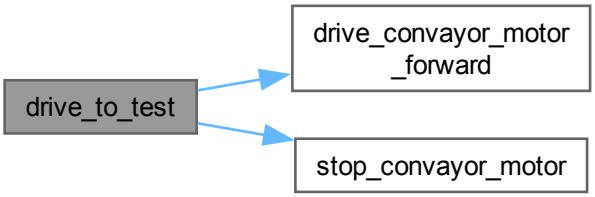
: void

Definition at line 112 of file ConveyorSensor.ino.

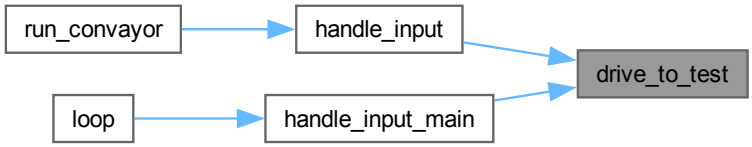
References drive_convayor_motor_forward(), sensor, and stop_convayor_motor().

Referenced by handle_input(), and handle_input_main().

Here is the call graph for this function:



Here is the caller graph for this function:



2.3.1.5 handle_input()

```
void handle_input (
            String input )
```

Remarks

: Handeling input for conveyor system

Parameters

	input

Returns

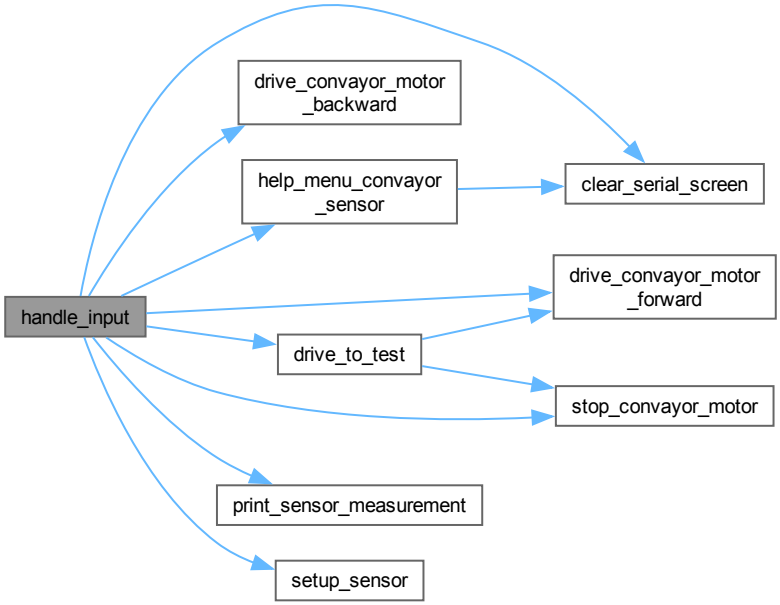
: void

Definition at line 165 of file ConveyorSensor.ino.

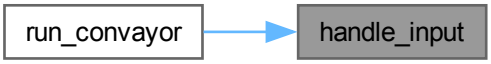
References clear_serial_screen(), drive_convayor_motor_backward(), drive_convayor_motor_forward(), drive_to_test(), help_menu_convayor_sensor(), print_sensor_measurement(), setup_sensor(), and stop_convayor_motor().

Referenced by run_convayor().

Here is the call graph for this function:



Here is the caller graph for this function:



2.3.1.6 help_menu_convayor_sensor()

```
void help_menu_convayor_sensor ( )
```

Remarks

: Printing info for the conveyor system

Returns

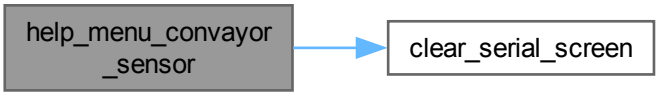
: void

Definition at line 130 of file ConveyorSensor.ino.

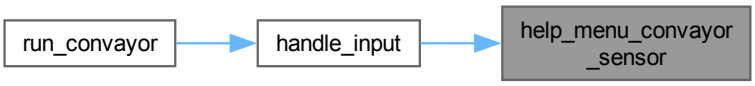
References clear_serial_screen().

Referenced by handle_input().

Here is the call graph for this function:



Here is the caller graph for this function:



2.3.1.7 print_sensor_measurement()

```
void print_sensor_measurement ( )
```

Remarks

: Printing the sensors measurement

Returns

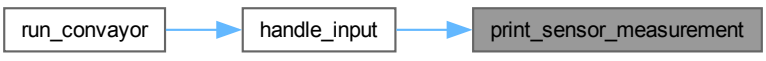
: void

Definition at line 149 of file ConveyorSensor.ino.

References sensor.

Referenced by handle_input().

Here is the caller graph for this function:



2.3.1.8 run_convayor()

```
void run_convayor ( )
```

Remarks

: Runs the conveyor system

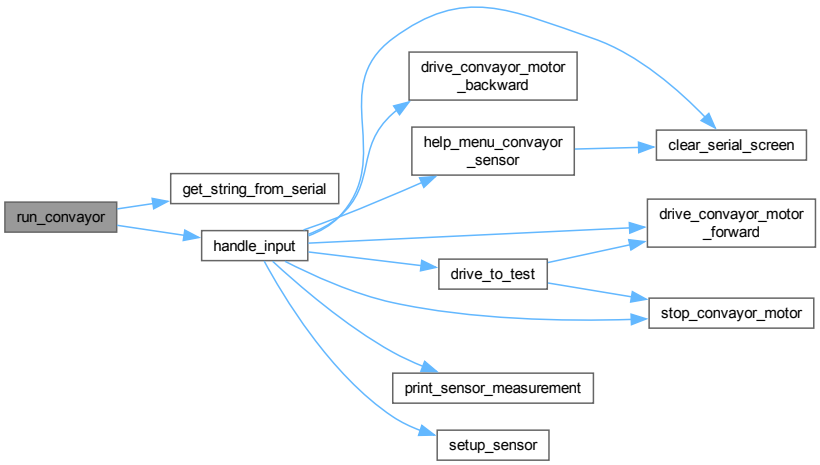
Returns

: void

Definition at line 190 of file ConveyorSensor.ino.

References get_string_from_serial(), and handle_input().

Here is the call graph for this function:



2.3.1.9 set_measurement_time_budget()

```
uint16_t set_measurement_time_budget ( )
```

Remarks

: Gets measurement time budget

Returns

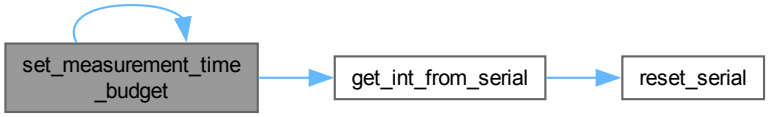
: measurement time budget

Definition at line 24 of file ConveyorSensor.ino.

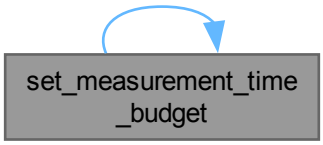
References get_int_from_serial(), and set_measurement_time_budget().

Referenced by set_measurement_time_budget().

Here is the call graph for this function:



Here is the caller graph for this function:



2.3.1.10 set_normal_range()

```
uint16_t set_normal_range ( )
```

Remarks

: Measureing the curent range

Returns

: millimeters

Definition at line 53 of file ConveyorSensor.ino.

References sensor.

2.3.1.11 set_timeout_sensor()

```
uint16_t set_timeout_sensor ( )
```

Remarks

: Show info about setting timeout of sensor

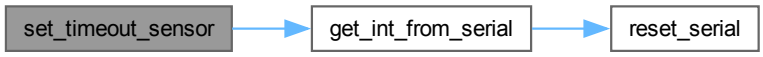
Returns

: time

Definition at line 14 of file ConveyorSensor.ino.

References get_int_from_serial().

Here is the call graph for this function:



2.3.1.12 setup_motor_driver()

```
void setup_motor_driver ( )
```

Remarks

: Setting up the motor for conveyor

Returns

: void

Definition at line 73 of file ConveyorSensor.ino.

References convayor_motor_minus, and convayor_motor_plus.

Referenced by setup().

Here is the caller graph for this function:

2.3.1.13 setup_sensor()

```
void setup_sensor ( )
```

Remarks

: Setting up the sensor

Returns

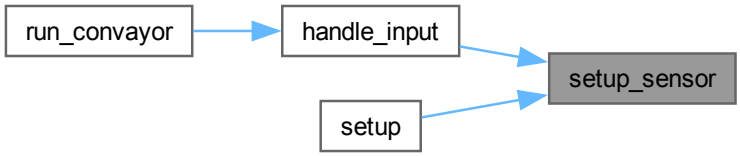
: void

Definition at line 40 of file ConveyorSensor.ino.

References sensor.

Referenced by handle_input(), and setup().

Here is the caller graph for this function:



2.3.1.14 stop_convayor_motor()

```
void stop_convayor_motor ( )
```

Remarks

: Stops the conveyor

Returns

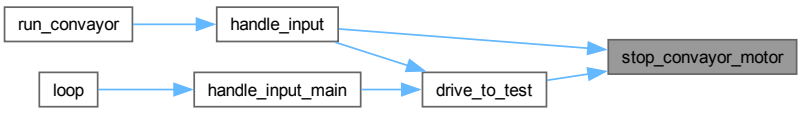
: void

Definition at line 100 of file ConveyorSensor.ino.

References convayor_motor_minus, and convayor_motor_plus.

Referenced by drive_to_test(), and handle_input().

Here is the caller graph for this function:



2.3.2 Variable Documentation

2.3.2.1 sensor

VL53L0X sensor

Definition at line 8 of file ConveyorSensor.ino.

Referenced by check_if_test_is_in_front(), drive_to_test(), print_sensor_measurement(), set_normal_range(), and setup_sensor().

2.4 ConveyorSensor.ino

Go to the documentation of this file.

```
00001 #include <VL53L0X.h>
00002 #include "ConveyorSensor.h"
00003 #include "Serial.h"
00004
00005
00006 /***********************************Sensor*********************************/
00007
00008 VL53L0X sensor;
00009
00014 uint16_t set_timeout_sensor(){
00015     Serial.println("Type The timeout in milliseconds (0 will disable timeout)");
00016     Serial.print("Enter number: ");
00017     return get_int_from_serial();
00018 }
00019
00024 uint16_t set_measurement_time_budget(){
00025     Serial.println("Type the measurement time budget in milliseconds (longer time allows for more
      accuret measurements)");
00026     Serial.print("Enter number: ");
00027     int measurement_time_budget = get_int_from_serial();
00028     if(measurement_time_budget < 20){
00029         Serial.println("Error measurement timing budget can't be lower than 20 ms");
00030         Serial.println("Try again");
00031         set_measurement_time_budget();
00032     }
00033     return measurement_time_budget;
00034 }
00035
00040 void setup_sensor(){
00041     sensor.setTimeout(0);
00042     if(!sensor.init()){
00043         Serial.println("Failed to detect and initialize sensor");
00044             while(1){}
00045     }
00046     sensor.setMeasurementTimingBudget(20000);
00047 }
00048
00053 uint16_t set_normal_range(){
00054     return sensor.readRangeSingleMillimeters();
00055 }
00056
00061 bool check_if_test_is_in_front(uint16_t normal_range){
00062     if(sensor.readRangeSingleMillimeters() < normal_range){
00063         return true;
00064     }
00065     return false;
00066 }
00067
00068 /*****************************Convayor motor************************************************/
00073 void setup_motor_driver(){
00074     pinMode(convayor_motor_plus, OUTPUT);
00075     pinMode(convayor_motor_minus, OUTPUT);
00076 }
00077
00082 void drive_convayor_motor_forward(){
00083     digitalWrite(convayor_motor_plus, LOW);
00084     digitalWrite(convayor_motor_minus, HIGH);
00085 }
00086
00091 void drive_convayor_motor_backward(){
00092     digitalWrite(convayor_motor_plus, HIGH);
```

```
00093     digitalWrite(convayor_motor_minus, LOW);
00094 }
00095
00100 void stop_convayor_motor(){
00101     digitalWrite(convayor_motor_plus, HIGH);
00102     digitalWrite(convayor_motor_minus, HIGH);
00103     delay(100);
00104     digitalWrite(convayor_motor_plus, LOW);
00105     digitalWrite(convayor_motor_minus, LOW);
00106 }
00107
00112 void drive_to_test(){
00113     drive_convayor_motor_forward();
00114     while(sensor.readRangeSingleMillimeters() < 100)
00115         delay(10);
00116     while(sensor.readRangeSingleMillimeters() > 100)
00117         delay(10);
00118     stop_convayor_motor();
00119     Serial.print("Done");
00120 }
00121
00122 /*-------------------------------------------------------------------------------------------*/
00123 // Serial
00124
00125
00130 void help_menu_convayor_sensor(){
00131     clear_serial_screen();
00132     Serial.println("The Convayor sensor measure when the patch is in front of it");
00133     Serial.println("-------------------------------------------------------------");
00134     Serial.println("To get back press 'b'");
00135     Serial.println("To clear the screen press 'c'");
00136     Serial.println("To set up the sensor press 'setup'");
00137     Serial.println("To read the measured distance press 'rd'");
00138     Serial.println("To drive to next test press 'next'");
00139     Serial.println("To drive motor forward press 'f'");
00140     Serial.println("To drive motor backward press 'b'");
00141     Serial.println("To stop motor press 's'");
00142 }
00143
00144
00149 void print_sensor_measurement(){
00150     uint16_t distance = sensor.readRangeSingleMillimeters();
00151     if(sensor.timeoutOccurred()){
00152         Serial.println("Error Timeout");
00153         return;
00154     }
00155     Serial.print("The distance is ");
00156     Serial.print(sensor.readRangeSingleMillimeters());
00157     Serial.print("mm\n");
00158 }
00159
00165 void handle_input(String input){
00166     if(input == "h")
00167         help_menu_convayor_sensor();
00168     else if(input == "c")
00169         clear_serial_screen();
00170     else if(input == "b")
00171         return;
00172     else if (input == "setup")
00173         setup_sensor();
00174     else if(input == "rd")
00175         print_sensor_measurement();
00176     else if(input == "next")
00177         drive_to_test();
00178     else if(input == "f")
00179         drive_convayor_motor_forward();
00180     else if(input == "b")
00181         drive_convayor_motor_backward();
00182     else if(input == "s")
00183         stop_convayor_motor();
00184 }
00185
00190 void run_convayor(){
00191     handle_input(get_string_from_serial());
00192 }
00193
```

2.5 ExternalSoftware_0.2.ino File Reference

```
#include <Wire.h>
#include "ConveyorSensor.h"
```

```
#include "Serial.h"
```

Functions

- void setup ()
- void loop ()

2.5.1 Function Documentation

2.5.1.1 loop()

```
void loop ( )
```

Remarks

: Loop

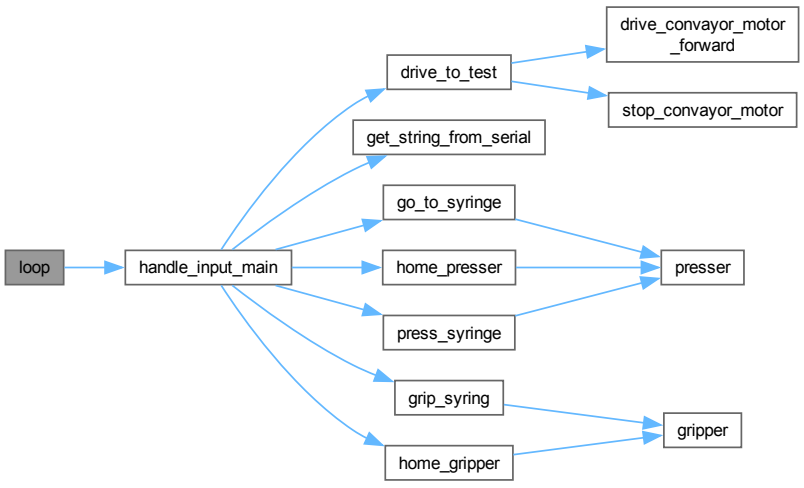
Returns

: void

Definition at line 22 of file ExternalSoftware_0.2.ino.

References handle_input_main().

Here is the call graph for this function:



2.5.1.2 setup()

```
void setup ( )
```

Remarks

: Setup

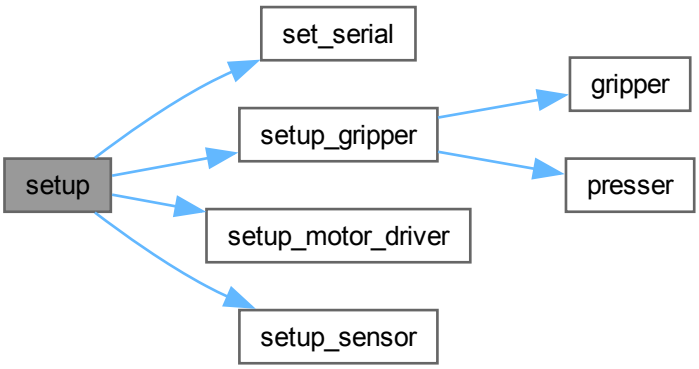
Returns

: void

Definition at line 10 of file ExternalSoftware_0.2.ino.

References set_serial(), setup_gripper(), setup_motor_driver(), and setup_sensor().

Here is the call graph for this function:



2.6 ExternalSoftware_0.2.ino

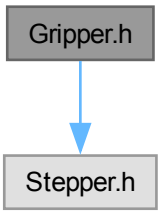
Go to the documentation of this file.

```
00001 #include <Wire.h>
00002 #include "ConveyorSensor.h"
00003 #include "Serial.h"
00004 //#include "Gripper.h"
00005
00010 void setup() {
00011     Wire.begin();
00012     set_serial();
00013     setup_gripper();
00014     setup_motor_driver();
00015     setup_sensor();
00016 }
00017
00022 void loop() {
00023     delay(100);
00024     //gripper_function();
00025     //run_convayor();
00026     handle_input_main();
00027     Serial.flush();
00028 }
```

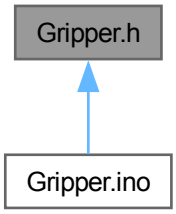
2.7 Gripper.h File Reference

```
#include <Stepper.h>
```

Include dependency graph for Gripper.h:



This graph shows which files directly or indirectly include this file:



Functions

- Stepper gripper (steps_per_revolution, 10, 11, 12, 13)
- Stepper presser (steps_per_revolution, 2, 3, 4, 5)
- void setup_gripper ()
- void home_gripper ()
- void home_presser ()
- void go_to_syringe ()
- void press_syringe (uint16_t)
- void grip_syring ()
- void set_motor_speed ()
- void run_stepper_for_gripping ()
- void run_stepper_for_presser (uint16_t)
- void gripper_function ()

Variables

- uint8_t switch_pin_presser = A2
- uint8_t switch_pin_gripper = A1
- uint8_t switch_pin_presser_syringe = A0
- const uint8_t steps_per_revolution = 200

2.7.1 Function Documentation

2.7.1.1 go_to_syringe()

```
void go_to_syringe ( )
```

Remarks

: Getting the presser down to the top of the syringe

Returns

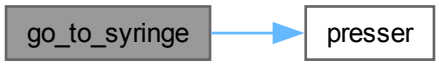
: void

Definition at line 80 of file Gripper.ino.

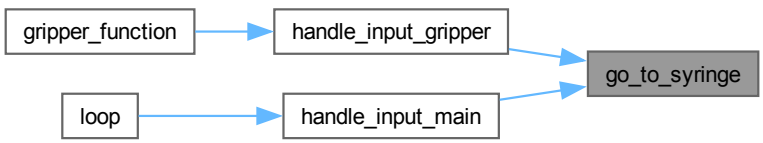
References presser(), and switch_pin_presser_syringe.

Referenced by handle_input_gripper(), and handle_input_main().

Here is the call graph for this function:



Here is the caller graph for this function:



2.7.1.2 grip_syring()

```
void grip_syring ( )
```

Remarks

: Gripping the syringe

Returns

: void

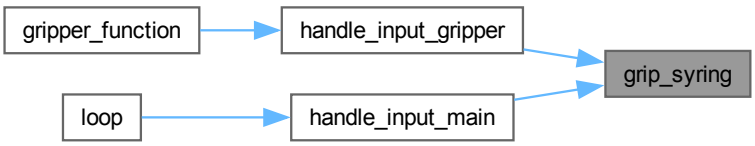
Definition at line 107 of file Gripper.ino.

References gripper().

Referenced by handle_input_gripper(), and handle_input_main().

Here is the call graph for this function:

Here is the caller graph for this function:

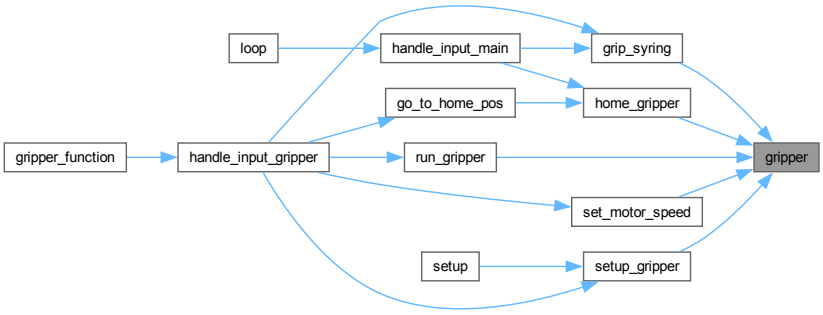


2.7.1.3 gripper()

```
Stepper gripper (
            steps_per_revolution ,
            10 ,
            11 ,
            12 ,
            13  )  [extern]
```

Referenced by grip_syring(), home_gripper(), run_gripper(), set_motor_speed(), and setup_gripper().

Here is the caller graph for this function:



2.7.1.4 gripper_function()

```
void gripper_function ( )
```

Definition at line 218 of file Gripper.ino.

References get_string_from_serial(), and handle_input_gripper().

Here is the call graph for this function:

2.7.1.5 home_gripper()

```
void home_gripper ( )
```

Remarks

: Setting the gripper to the home position

Returns

: void

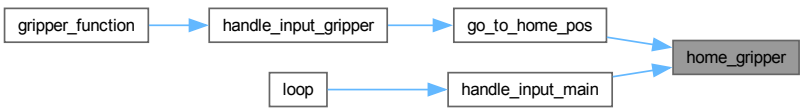
Definition at line 32 of file Gripper.ino.

References gripper(), and switch_pin_gripper.

Referenced by go_to_home_pos(), and handle_input_main().

Here is the call graph for this function:

Here is the caller graph for this function:



2.7.1.6 home_presser()

```
void home_presser ( )
```

Remarks

: Setting the presser to the home position

Returns

: void

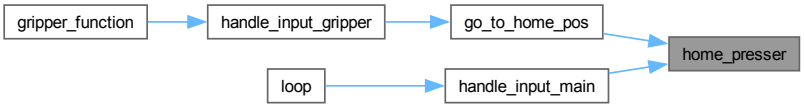
Definition at line 45 of file Gripper.ino.

References presser(), and switch_pin_presser.

Referenced by go_to_home_pos(), and handle_input_main().

Here is the call graph for this function:

Here is the caller graph for this function:



2.7.1.7 press_syringe()

```
void press_syringe (
            uint16_t micro_liter )
```

Remarks

: Pressing the number of micro litters out

Parameters

	micro litter

Returns

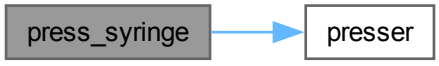
: void

Definition at line 95 of file Gripper.ino.

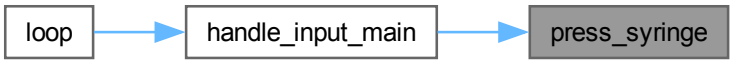
References presser().

Referenced by handle_input_main().

Here is the call graph for this function:



Here is the caller graph for this function:

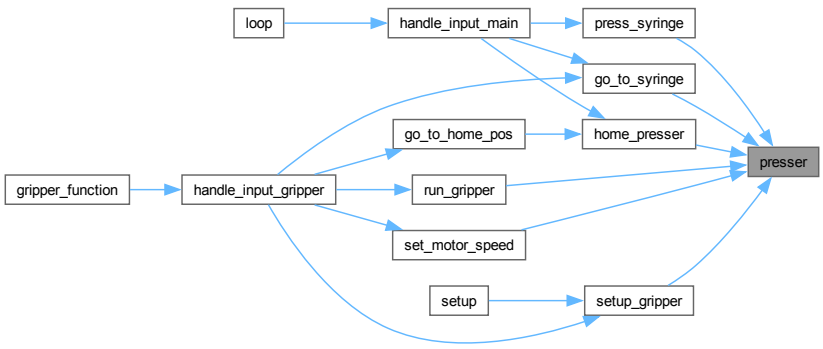


2.7.1.8 presser()

```
Stepper presser (
            steps_per_revolution ,
            2 ,
            3 ,
            4 ,
            5  )  [extern]
```

Referenced by go_to_syringe(), home_presser(), press_syringe(), run_gripper(), set_motor_speed(), and setup_gripper().

Here is the caller graph for this function:



2.7.1.9 run_stepper_for_gripping()

```
void run_stepper_for_gripping ( )
```

2.7.1.10 run_stepper_for_presser()

```
void run_stepper_for_presser (
            uint16_t  )
```

2.7.1.11 set_motor_speed()

```
void set_motor_speed ( )
```

Remarks

: Getting an input and setting the speed on the correct stepper

Returns

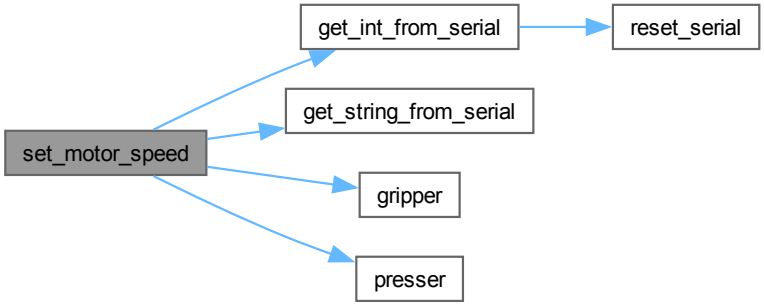
: void

Definition at line 118 of file Gripper.ino.

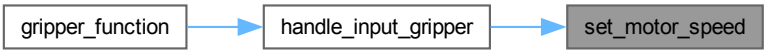
References get_int_from_serial(), get_string_from_serial(), gripper(), and presser().

Referenced by handle_input_gripper().

Here is the call graph for this function:



Here is the caller graph for this function:



2.7.1.12 setup_gripper()

```
void setup_gripper ( )
```

Remarks

: beskrivelse

Parameters

	parameter
	parameter2

Returns

: hvad der retuneres

Remarks

: Setting up the gripper and sets the default speed for the stepper stepper

Returns

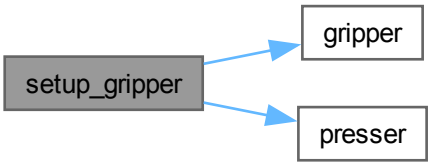
: void

Definition at line 17 of file Gripper.ino.

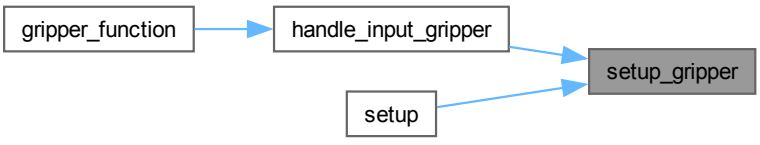
References gripper(), presser(), switch_pin_gripper, switch_pin_presser, and switch_pin_presser_syringe.

Referenced by handle_input_gripper(), and setup().

Here is the call graph for this function:



Here is the caller graph for this function:



2.7.2 Variable Documentation

2.7.2.1 steps_per_revolution

```
const uint8_t steps_per_revolution = 200  [extern]
```

Referenced by get_num_of_steps().

2.7.2.2 switch_pin_gripper

```
uint8_t switch_pin_gripper = A1  [extern]
```

Referenced by home_gripper(), and setup_gripper().

2.7.2.3 switch_pin_presser

```
uint8_t switch_pin_presser = A2  [extern]
```

Referenced by home_presser(), and setup_gripper().

2.7.2.4 switch_pin_presser_syringe

```
uint8_t switch_pin_presser_syringe = A0  [extern]
```

Referenced by go_to_syringe(), and setup_gripper().

2.8 Gripper.h

Go to the documentation of this file.

```
00001 #ifndef GRIPPER_H
00002 #define GRIPPER_H
00003 #include <Stepper.h>
00004
00005 extern uint8_t switch_pin_presser = A2;
00006 extern uint8_t switch_pin_gripper = A1;
00007 extern uint8_t switch_pin_presser_syringe = A0;
00008
00009 extern const uint8_t steps_per_revolution = 200;
00010
00011 // the gripping mecanisem is conected to pin 8, 9, 10, 11
00012 extern Stepper gripper(steps_per_revolution, 10, 11, 12, 13);
00013
00014 // the pressing mecanisem is conected to pin 2, 3, 4, 5
00015 extern Stepper presser(steps_per_revolution, 2, 3, 4, 5);
00016
00017
00018 void setup_gripper();
00019 void home_gripper();
00020 void home_presser();
00021 void go_to_syringe();
00022 void press_syringe(uint16_t);
00023 void grip_syring();
00024 void set_motor_speed();
00025 void run_stepper_for_gripping();
00026 void run_stepper_for_presser(uint16_t);
00027 void gripper_function();
00028
00029 #endif
```

2.9 Gripper.ino File Reference

```
#include "Gripper.h"
#include "Serial.h"
```

Functions

- void setup_gripper ()
- void home_gripper ()
- void home_presser ()
- void go_to_home_pos ()
- void go_to_syringe ()
- void press_syringe (uint16_t micro_liter)
- void grip_syring ()
- void set_motor_speed ()
- uint16_t get_num_of_steps ()
- void run_gripper ()
- void help_menu_gripper ()
- void handle_input_gripper (String input)
- void gripper_function ()

2.9.1 Function Documentation

2.9.1.1 get_num_of_steps()

```
uint16_t get_num_of_steps ( )
```

Remarks

: Getting number of turns as input and returning number of steps

Returns

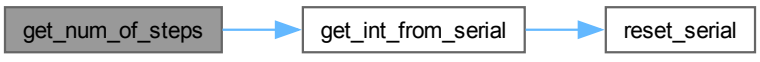
: Number of steps

Definition at line 143 of file Gripper.ino.

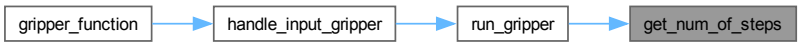
References get_int_from_serial(), and steps_per_revolution.

Referenced by run_gripper().

Here is the call graph for this function:



Here is the caller graph for this function:



2.9.1.2 go_to_home_pos()

```
void go_to_home_pos ( )
```

Remarks

: Getting g or p as a input and calling the correct function

Returns

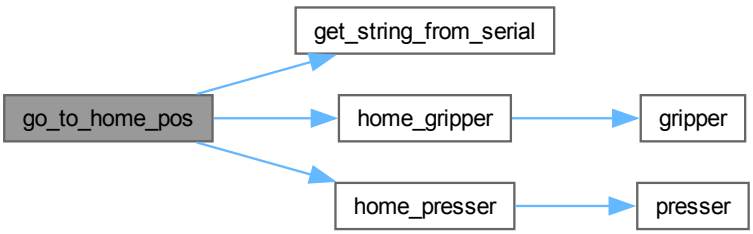
: void

Definition at line 60 of file Gripper.ino.

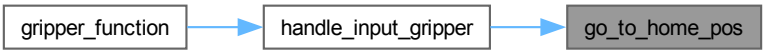
References get_string_from_serial(), home_gripper(), and home_presser().

Referenced by handle_input_gripper().

Here is the call graph for this function:



Here is the caller graph for this function:



2.9.1.3 go_to_syringe()

```
void go_to_syringe ( )
```

Remarks

: Getting the presser down to the top of the syringe

Returns

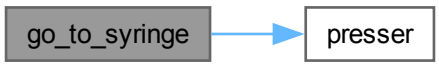
: void

Definition at line 80 of file Gripper.ino.

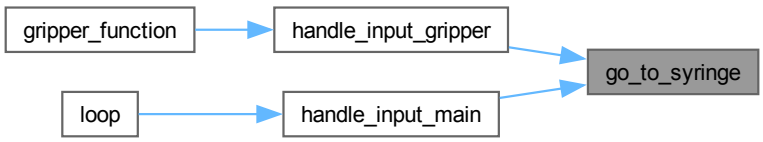
References presser(), and switch_pin_presser_syringe.

Referenced by handle_input_gripper(), and handle_input_main().

Here is the call graph for this function:



Here is the caller graph for this function:



2.9.1.4 grip_syring()

```
void grip_syring ( )
```

Remarks

: Gripping the syringe

Returns

: void

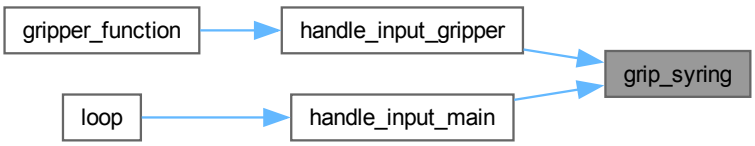
Definition at line 107 of file Gripper.ino.

References gripper().

Referenced by handle_input_gripper(), and handle_input_main().

Here is the call graph for this function:

Here is the caller graph for this function:



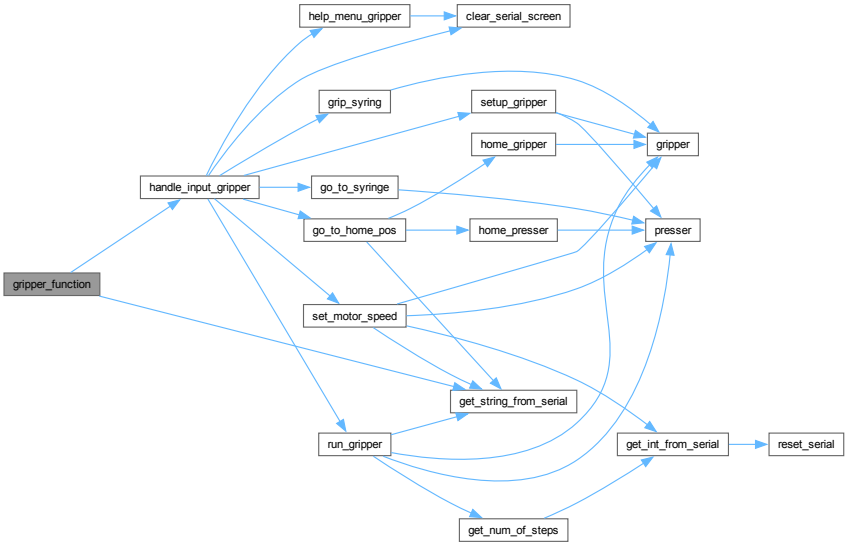
2.9.1.5 gripper_function()

```
void gripper_function ( )
```

Definition at line 218 of file Gripper.ino.

References get_string_from_serial(), and handle_input_gripper().

Here is the call graph for this function:



2.9.1.6 handle_input_gripper()

```
void handle_input_gripper (
            String input )
```

Remarks

: Handle input for gripper

Parameters

	input

Returns

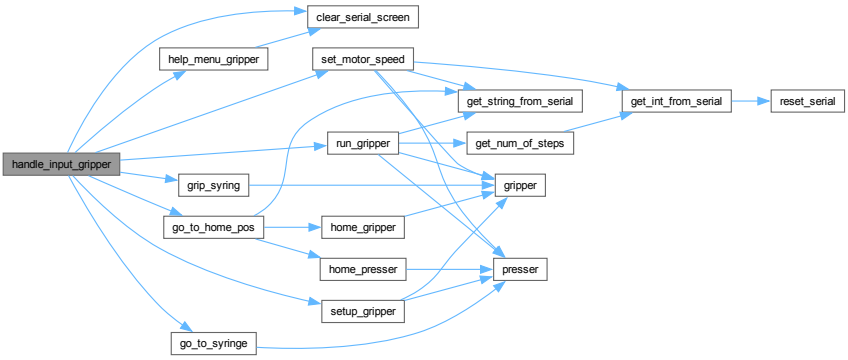
: void

Definition at line 194 of file Gripper.ino.

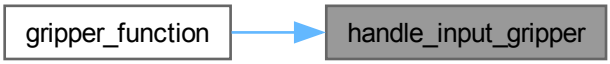
References clear_serial_screen(), go_to_home_pos(), go_to_syringe(), grip_syring(), help_menu_gripper(), run_gripper(), set_motor_speed(), and setup_gripper().

Referenced by gripper_function().

Here is the call graph for this function:



Here is the caller graph for this function:



2.9.1.7 help_menu_gripper()

```
void help_menu_gripper ( )
```

Remarks

: Printing info to Serial

Returns

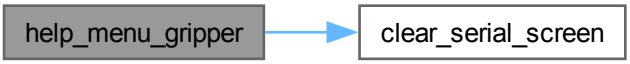
: void

Definition at line 173 of file Gripper.ino.

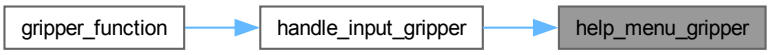
References clear_serial_screen().

Referenced by handle_input_gripper().

Here is the call graph for this function:



Here is the caller graph for this function:



2.9.1.8 home_gripper()

```
void home_gripper ( )
```

Remarks

: Setting the gripper to the home position

Returns

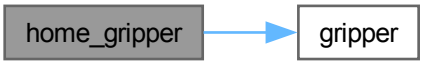
: void

Definition at line 32 of file Gripper.ino.

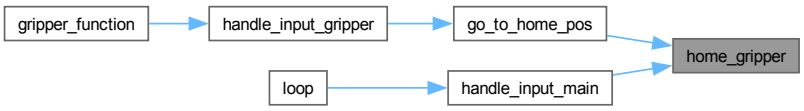
References gripper(), and switch_pin_gripper.

Referenced by go_to_home_pos(), and handle_input_main().

Here is the call graph for this function:



Here is the caller graph for this function:



2.9.1.9 home_presser()

```
void home_presser ( )
```

Remarks

: Setting the presser to the home position

Returns

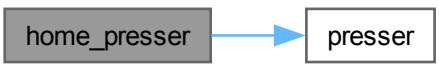
: void

Definition at line 45 of file Gripper.ino.

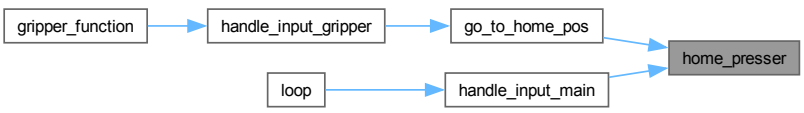
References presser(), and switch_pin_presser.

Referenced by go_to_home_pos(), and handle_input_main().

Here is the call graph for this function:



Here is the caller graph for this function:



2.9.1.10 press_syringe()

```
void press_syringe (
            uint16_t micro_liter )
```

Remarks

: Pressing the number of micro litters out

Parameters

	micro litter

Returns

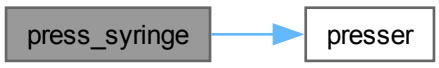
: void

Definition at line 95 of file Gripper.ino.

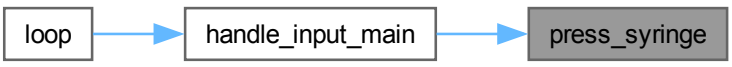
References presser().

Referenced by handle_input_main().

Here is the call graph for this function:



Here is the caller graph for this function:



2.9.1.11 run_gripper()

```
void run_gripper ( )
```

Remarks

: Getting g or p as an input and turning the coresponding stepper

Returns

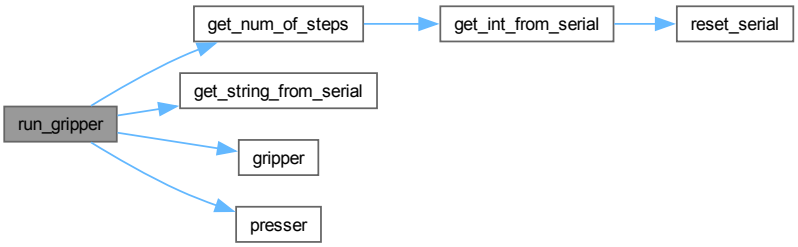
: void

Definition at line 153 of file Gripper.ino.

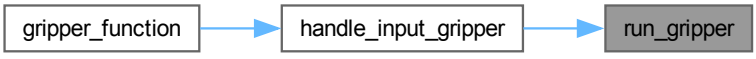
References get_num_of_steps(), get_string_from_serial(), gripper(), and presser().

Referenced by handle_input_gripper().

Here is the call graph for this function:



Here is the caller graph for this function:



2.9.1.12 set_motor_speed()

```
void set_motor_speed ( )
```

Remarks

: Getting an input and setting the speed on the correct stepper

Returns

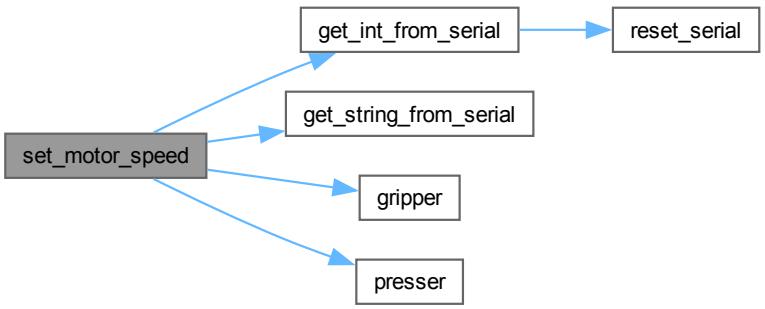
: void

Definition at line 118 of file Gripper.ino.

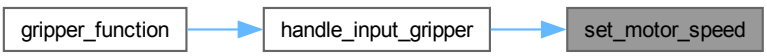
References get_int_from_serial(), get_string_from_serial(), gripper(), and presser().

Referenced by handle_input_gripper().

Here is the call graph for this function:



Here is the caller graph for this function:



2.9.1.13 setup_gripper()

```
void setup_gripper ( )
```

Remarks

: beskrivelse

Parameters

	parameter
	parameter2

Returns

: hvad der retuneres

Remarks

: Setting up the gripper and sets the default speed for the stepper stepper

Returns

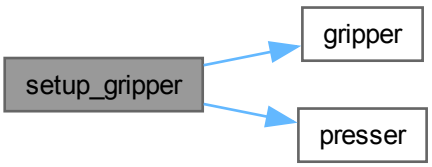
: void

Definition at line 17 of file Gripper.ino.

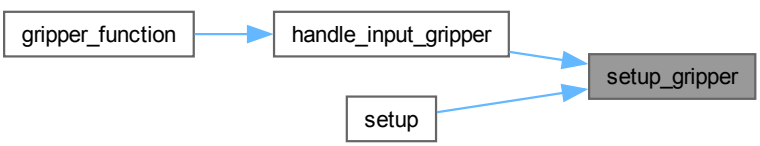
References gripper(), presser(), switch_pin_gripper, switch_pin_presser, and switch_pin_presser_syringe.

Referenced by handle_input_gripper(), and setup().

Here is the call graph for this function:



Here is the caller graph for this function:



2.10 Gripper.ino

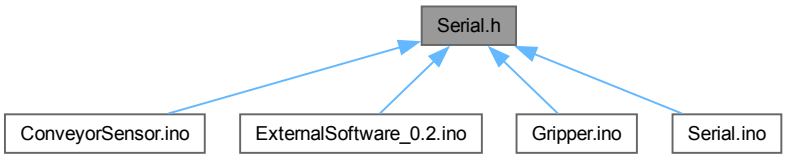
Go to the documentation of this file.

```
00001 #include "Gripper.h"
00002 #include "Serial.h"
00003
00004 // 2.28 steps per micro litter
00005
00017 void setup_gripper(){
00018     pinMode(switch_pin_presser, INPUT);
00019     pinMode(switch_pin_gripper, INPUT);
00020     pinMode(switch_pin_presser_syringe, INPUT);
00021     // Setting the default speed
00022     gripper.setSpeed(100);
00023     presser.setSpeed(100);
00024 }
00025
00026 /**************************Home*****************************/
00027
00032 void home_gripper(){
00033     uint8_t contact = digitalRead(switch_pin_gripper);
00034     while(contact != 1){
00035         gripper.step(1);
00036         contact = digitalRead(switch_pin_gripper);
00037     }
00038     Serial.print("Done");
00039 }
00040
00045 void home_presser(){
00046     presser.setSpeed(100);
00047     uint8_t contact = digitalRead(switch_pin_presser);
00048     while(contact != 0){
00049         presser.step(1);
00050         contact = digitalRead(switch_pin_presser);
00051     }
00052     Serial.print("Done");
00053 }
00054
00055
00060 void go_to_home_pos(){
00061     Serial.println("To home the gripper press 'g'");
00062     Serial.println("To home the presser press 'p'");
00063     String input = get_string_from_serial();
00064     if(input == "g"){
00065         home_gripper();
00066     }
00067     else if(input == "p"){
00068         home_presser();
00069     }
00070     else{
00071         Serial.println("Input not valid");
00072     }
00073 }
00074
00075 /**************************Pressing*****************************/
00080 void go_to_syringe(){
00081     presser.setSpeed(100);
00082     uint8_t contact = digitalRead(switch_pin_presser_syringe);
00083     while(contact != 0){
00084         presser.step(-1);
00085         contact = digitalRead(switch_pin_presser_syringe);
00086     }
00087     Serial.print("Done");
00088 }
00089
00095 void press_syringe(uint16_t micro_liter){
00096     presser.setSpeed(50);
00097     uint16_t step_to_take = micro_liter * 2.28;
00098     presser.step(-step_to_take);
00099     Serial.print("Done");
00100 }
00101
00102
00107 void grip_syring(){
00108     gripper.step(-680);
00109     Serial.print("Done");
00110 }
00111
00112 /**************************Config*****************************/
00113
00118 void set_motor_speed(){
00119     Serial.println("To set the speed for the gripping press 'g'");
00120     Serial.println("To set the speed for the pressing press 'p'");
00121     Serial.println("To set the speed for them both press 'b'");
00122     String input = get_string_from_serial();
```

```
00123     Serial.print("Type the speed: ");
00124     uint8_t speed = get_int_from_serial();
00125     if(input = "b"){
00126         gripper.setSpeed(speed);
00127         presser.setSpeed(speed);
00128     }
00129     else if(input = "g"){
00130         gripper.setSpeed(speed);
00131     }
00132     else if(input = "p"){
00133         presser.setSpeed(speed);
00134     }
00135 }
00136
00137 /**********************Debug*************************/
00138
00143 uint16_t get_num_of_steps(){
00144     Serial.print("Type the number of turns to turn the stepper: ");
00145     uint16_t turns = get_int_from_serial();
00146     return turns*steps_per_revolution;
00147 }
00148
00153 void run_gripper(){
00154     Serial.println("To run the gripping mecanisem press 'g'");
00155     Serial.println("To run the pressing mecanisem press 'p'");
00156     String input = get_string_from_serial();
00157     int16_t num_steps_to_turn = get_num_of_steps();
00158     if(input == "g"){
00159         Serial.println("hi");
00160         gripper.step(-num_steps_to_turn);
00161     }
00162     else if(input == "p"){
00163         presser.step(-num_steps_to_turn);
00164     }
00165 }
00166
00167 /************************Info screens************************/
00168
00173 void help_menu_gripper(){
00174     clear_serial_screen();
00175     Serial.println("There are to stepper motors on the gripper,");
00176     Serial.println("there are one for gripping the syringe,");
00177     Serial.println("and one for pressing on the syringe.");
00178     Serial.println("-------------------------------------------");
00179     Serial.println("To get back press 'b'");
00180     Serial.println("To clear the screen press 'c'");
00181     Serial.println("To setup the gripper press 'setup'");
00182     Serial.println("To home the gripper press 'home'");
00183     Serial.println("To get presser to syringe press 'ps'");
00184     Serial.println("To grip the syringe press 'grip'");
00185     Serial.println("To set the speed press 'sp'");
00186     Serial.println("To run the gripper pess 'run'");
00187 }
00188
00194 void handle_input_gripper(String input){
00195     //Serial.print("input was: ");
00196     if(input == "b")
00197         return;
00198     else if(input == "c")
00199         clear_serial_screen();
00200     else if(input == "h")
00201         help_menu_gripper();
00202     else if(input == "setup")
00203         setup_gripper();
00204     else if(input == "home")
00205         go_to_home_pos();
00206     else if(input == "ps")
00207         go_to_syringe();
00208     else if(input == "grip")
00209         grip_syring();
00210     else if(input == "sp"){
00211         set_motor_speed();
00212     }
00213     else if(input == "run"){
00214         run_gripper();
00215     }
00216 }
00217
00218 void gripper_function(){
00219         handle_input_gripper(get_string_from_serial());
00220 }
```

2.11 Serial.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void set_serial ()
- void reset_serial ()
- void clear_serial_screen ()
- uint16_t get_int_from_serial ()
- String get_string_from_serial ()

2.11.1 Function Documentation

2.11.1.1 clear_serial_screen()

```
void clear_serial_screen ( )
```

Remarks

: Clearing the serial screen

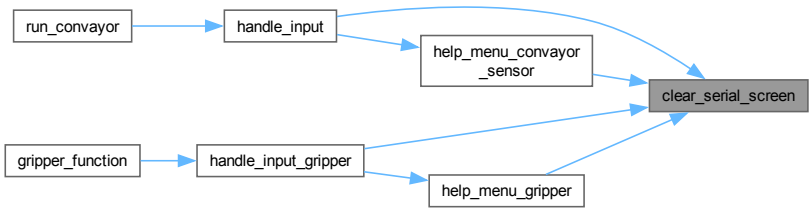
Returns

: void

Definition at line 26 of file Serial.ino.

Referenced by handle_input(), handle_input_gripper(), help_menu_convayor_sensor(), and help_menu_gripper().

Here is the caller graph for this function:



2.11.1.2 get_int_from_serial()

```
uint16_t get_int_from_serial ( )
```

Remarks

: Getting an int from serial and returns it

Returns

: input

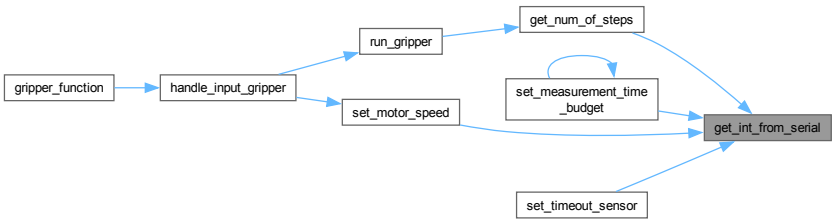
Definition at line 36 of file Serial.ino.

References reset_serial().

Referenced by get_num_of_steps(), set_measurement_time_budget(), set_motor_speed(), and set_timeout_sensor().

Here is the call graph for this function:

Here is the caller graph for this function:



2.11.1.3 get_string_from_serial()

```
String get_string_from_serial ( )
```

Remarks

: Getting a string from serial and returns it

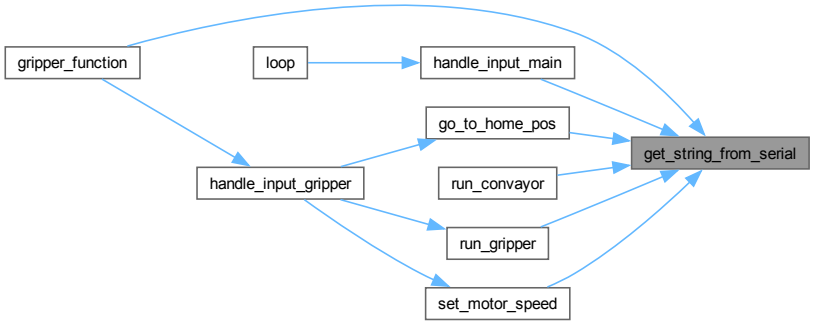
Returns

: input

Definition at line 53 of file Serial.ino.

Referenced by go_to_home_pos(), gripper_function(), handle_input_main(), run_convayor(), run_gripper(), and set_motor_speed().

Here is the caller graph for this function:



2.11.1.4 reset_serial()

```
void reset_serial ( )
```

Remarks

: Resets the Serial

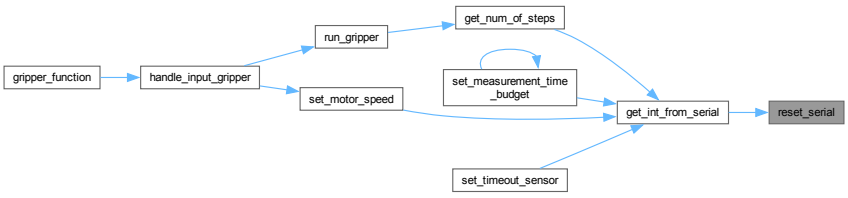
Returns

: void

Definition at line 17 of file Serial.ino.

Referenced by get_int_from_serial().

Here is the caller graph for this function:



2.11.1.5 set_serial()

```
void set_serial ( )
```

Remarks

: Setting up the Serial

Returns

: void

Definition at line 7 of file Serial.ino.

2.12 Serial.h

Go to the documentation of this file.

```
00001 #ifndef SERIAL_H
00002 #define SERIAL_H
00003
00004 void set_serial();
00005 void reset_serial();
00006 void clear_serial_screen();
00007 uint16_t get_int_from_serial();
00008 String get_string_from_serial();
00009
00010 #endif SERIAL_H
```

2.13 Serial.ino File Reference

```
#include "Serial.h"
```

Functions

- void set_serial ()
- void reset_serial ()
- void clear_serial_screen ()
- uint16_t get_int_from_serial ()
- String get_string_from_serial ()
- void handle_input_main ()

2.13.1 Function Documentation

2.13.1.1 clear_serial_screen()

```
void clear_serial_screen ( )
```

Remarks

: Clearing the serial screen

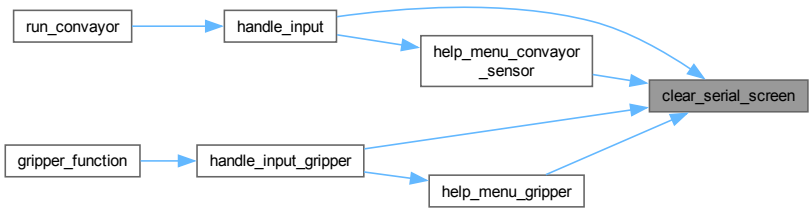
Returns

: void

Definition at line 26 of file Serial.ino.

Referenced by handle_input(), handle_input_gripper(), help_menu_convayor_sensor(), and help_menu_gripper().

Here is the caller graph for this function:



2.13.1.2 get_int_from_serial()

```
uint16_t get_int_from_serial ( )
```

Remarks

: Getting an int from serial and returns it

Returns

: input

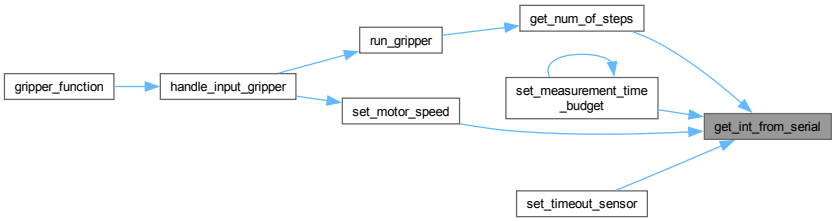
Definition at line 36 of file Serial.ino.

References reset_serial().

Referenced by get_num_of_steps(), set_measurement_time_budget(), set_motor_speed(), and set_timeout_sensor().

Here is the call graph for this function:

Here is the caller graph for this function:



2.13.1.3 get_string_from_serial()

```
String get_string_from_serial ( )
```

Remarks

: Getting a string from serial and returns it

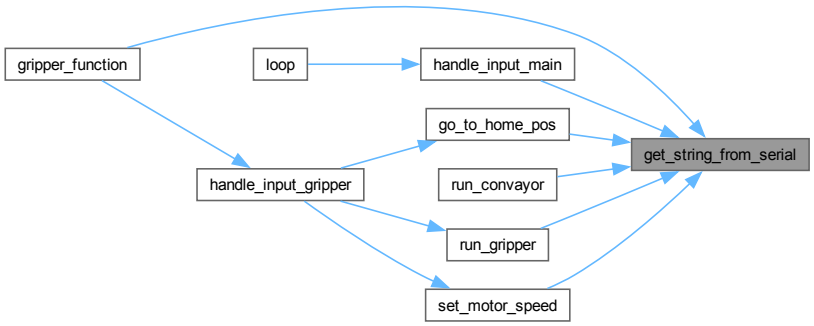
Returns

: input

Definition at line 53 of file Serial.ino.

Referenced by go_to_home_pos(), gripper_function(), handle_input_main(), run_convayor(), run_gripper(), and set_motor_speed().

Here is the caller graph for this function:



2.13.1.4 handle_input_main()

```
void handle_input_main ( )
```

Remarks

: Handle input for the main system

Returns

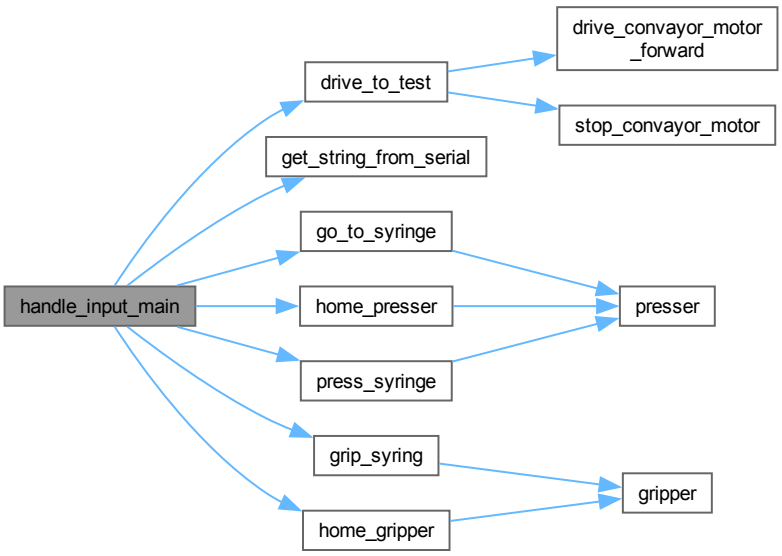
: void

Definition at line 67 of file Serial.ino.

References drive_to_test(), get_string_from_serial(), go_to_syringe(), grip_syring(), home_gripper(), home_presser(), and press_syringe().

Referenced by loop().

Here is the call graph for this function:



Here is the caller graph for this function:

2.13.1.5 reset_serial()

```
void reset_serial ( )
```

Remarks

: Resets the Serial

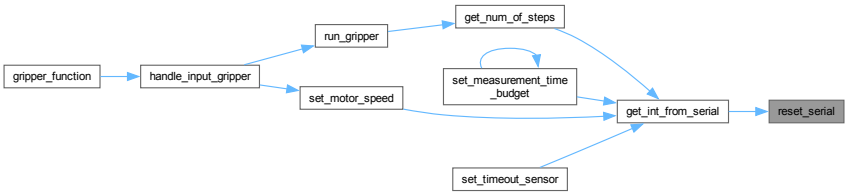
Returns

: void

Definition at line 17 of file Serial.ino.

Referenced by get_int_from_serial().

Here is the caller graph for this function:



2.13.1.6 set_serial()

```
void set_serial ( )
```

Remarks

: Setting up the Serial

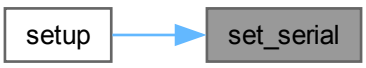
Returns

: void

Definition at line 7 of file Serial.ino.

Referenced by setup().

Here is the caller graph for this function:



2.14 Serial.ino

Go to the documentation of this file.

```
00001 #include "Serial.h"
00002
00007 void set_serial(){
00008     Serial.begin(9600);
00009     while(!Serial) delay(10);
00010     Serial.setTimeout(1000);
00011 }
00012
00017 void reset_serial(){
00018     Serial.end();
00019     Serial.begin(9600);
00020 }
00021
00026 void clear_serial_screen(){
00027     // prints 100 empty lines to clear screen
00028     for(int i=0; i<100; i++)
00029         Serial.println("");
00030 }
00031
00036 uint16_t get_int_from_serial(){
00037     int indput = 0;
00038     int timeOut = 0;
00039     while(indput < 1 && timeOut < 30){
00040         indput = Serial.parseInt(SKIP_ALL);
00041         timeOut++;
00042         delay(1000);
00043     }
00044     reset_serial();
00045     return indput;
00046 }
00047
00052 //bool readyForCommand = false;
00053 String get_string_from_serial(){
00054     //Serial.print("Enter Command: ");
00055     String input = "";
00056     while(input == ""){
00057         input = Serial.readString();
00058         input.trim();
00059     }
00060     return input;
00061 }
00062
00067 void handle_input_main(){
00068     String input_raw = get_string_from_serial();
00069     String input = "";
00070     uint8_t i = 0;
00071     char curen_char = input_raw.charAt(0);
00072     while(int(curen_char) != 45){
00073         input += curen_char;
00074         i++;
00075         if(i == input_raw.length())
00076             break;
00077         curen_char = input_raw.charAt(i);
00078     }
00079     i++;
00080     String input2 = "";
00081     for(;i<input_raw.length(); i++){
00082         input2 += input_raw.charAt(i);
00083     }
00084     if(input == "home "){
00085         if(input2 == "g")
00086             home_gripper();
00087         else
00088             home_presser();
00089     }
00090     else if(input == "close")
00091         grip_syring();
00092     else if(input == "prepare pressing")
00093         go_to_syringe();
00094     else if(input == "new test")
00095         drive_to_test();
00096     else if(input == "press ")
00097         press_syringe(input2.toInt());
00098 }
```

Index