Camduino

Generated by Doxygen 1.8.7

Wed Jul 23 2014 11:54:43

Contents

1 Main Page

Camduino is an interface to speak with Arduino over I2C. You can get a red ball position and state of proximity sensors. The main goal of this project is to supply an easy-to-use interface to communicate with Arduino as part of ImpRo project demonstrator, *Pongduino*.

I2C bus is also called Two Wire Interface (TWI) as in the Arduino reference/librairy.

Arduino setup

Arduino board must be connected to CMUcam4 and five presence sensors on pins 3 to 7 and run *CMU_Tracker_↩ I2C.ino*.

The wire

To connect an Arduino with a NXT, I made my own wire.

- white : ground
- black : ground
- red : analog
- green : 5V
- blue : SDA (Arduino pin A4)
- yellow : SCL (Arduino pin A5)

NXT can't supply enought energy for Arduino, so use an external source.

Trampoline settings

The Trampoline revision running on the NXT must use the custom I2C driver "from Armel". This driver is supplied with the Camduino library.

Modifying the driver at your own risk.

Arduino port and NXT

When calling `init_camduino(NXT_PORT_S#)` if you're **NOT** using **NXT_PORT_S4**, you should change it on the driver file i2c.c in *Trampolie_folder*/machines/arm/nxt/drivers/lejos_nxj/src/nxtvm/platform/nxt/i2c.c

Usage

- First, include `camduino.h` in your oil and C files.
- Initiate the driver library by calling `init_camduino(NXT_PORT_S4)`, eventualy, initiate other I2C devices and finally call `i2c_init()`.
- Then you get data from `Camduino`.
- Code and think hard. Good luck!

2 Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

3 File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

4 Data Structure Documentation

4.1 position Struct Reference

```
#include <camduino.h>
```

Data Fields

- int x
- int y

4.1.1 Detailed Description

store ball position

4.1.2 Field Documentation

4.1.2.1 int position::x

x coordinate of the ball

4.1.2.2 int position::y

y coordinate of the ball

The documentation for this struct was generated from the following file:

- camduino.h

5 File Documentation

5.1 camduino.c File Reference

```
#include "camduino.h"
#include "nxt_avr.h"
```

Functions

- void init_camduino (int i2c_port)
- void get_ball_position (struct position *ball)
- int object_detected ()
- int get_pstate (enum psensor sensor)

5.1.1 Detailed Description

5.1.2 File description

Librairy to get some data from an Arduino over I2C. CMUcam4 must be connect on it and Arduino must run "CM↩ U_Tracker_I2C" program, distance sensors must be connected to the board too.

5.1.3 Copyright

5.1.4 File informations

Date

2014/07/21

Author

Benjamin Sientzoff

Version

0.1

5.1.5 Function Documentation

5.1.5.1 void get_ball_position (struct **position** ∗ *ball*)

Fill a position strucuture to get ball position

Parameters

in	*ball*	A reference to the struct to fill

5.1.5.2 int get_pstate (enum **psensor** *sensor*)

Get an given presence sensor state

Parameters

in	*sensor*	Name of the presence sensor to examine
out	*int*	state of the sensor, return 0 if no detected object, else 1

5.1.5.3 void init_camduino (int *i2c_port*)

Initiate Arduino driver, don't forget to call i2c_init() after. Set up i2c port, tram communication and *power* supply.

Parameters

in	*i2c_port*	NXT port where Arduino is connected

5.1.5.4 int object_detected ( )

Get global presence sensors state

Parameters

out	*int*	State of the presence sensors, equals to NO_DETECTED_OBJECT if an object is not detected, else return a different value

5.2 camduino.h File Reference

```
#include "i2c.h"
```

Data Structures

- struct position

Macros

- #define NO_DETECTED_OBJECT 31

Enumerations

- enum psensor {
  PSENSOR_A = 0x1, PSENSOR_B = 0x2, PSENSOR_C = 0x4, PSENSOR_D = 0x8,
  PSENSOR_E = 0x10 }

Functions

- void init_camduino (int)
- void get_ball_position (struct position *)
- int object_detected ()
- int get_pstate (enum psensor)

Variables

- u8 i2c_data [I2C_PORT_N][I2C_DATA_N]
- int arduino_port
- int pstate = NO_DETECTED_OBJECT

5.2.1 Detailed Description

5.2.2 File description

Librairy to get some data from an Arduino over I2C. CMUcam4 must be connected on it and Arduino must run "CMU_Tracker_I2C" program, distance sensors must be connected to the board too.

5.2.3 Copyright

5.2.4 File informations

Date

2014/07/18

Author

Benjamin Sientzoff

Version

0.1

5.2.5 Macro Definition Documentation

5.2.5.1 #define NO_DETECTED_OBJECT 31

state of (all) presence sensors when there is no object

5.2.6 Enumeration Type Documentation

5.2.6.1 enum psensor

list of presence sensors connected to the Arduino board

Enumerator

PSENSOR_A the first presence sensor on the left

PSENSOR_B the second presence sensor on the left

PSENSOR_C presence sensor on the middle

PSENSOR_D the second sensor of the right

PSENSOR_E the last sensor (on the right)

5.2.7 Function Documentation

5.2.7.1 void get_ball_position (struct position * *ball*)

Fill a position strucuture to get ball position

Fill a position strucuture to get ball position

Parameters

in	*ball*	A reference to the struct to fill

5.2.7.2 int get_pstate (enum psensor *sensor*)

get an given presence sensor state

Get an given presence sensor state

Parameters

in	*sensor*	Name of the presence sensor to examine
out	*int*	state of the sensor, return 0 if no detected object, else 1

5.2.7.3 void init_camduino (int *i2c_port*)

Initiate Arduino driver

Initiate Arduino driver, don't forget to call i2c_init() after. Set up i2c port, tram communication and *power* supply.

Parameters

in	*i2c_port*	NXT port where Arduino is connected

5.2.7.4 int object_detected ( )

Get global presence sensors state, consider the five sensors as one

Get global presence sensors state

Parameters

out	*int*	State of the presence sensors, equals to NO_DETECTED_OBJECT if an object is not detected, else return a different value

5.2.8 Variable Documentation

5.2.8.1 int arduino_port

NXT port connected to Arduino board

5.2.8.2 u8 i2c_data[I2C_PORT_N][I2C_DATA_N]

buffers storing I2C data

5.2.8.3 int pstate = NO_DETECTED_OBJECT

global state of presence sensors

Index