|  |
| --- |
| UKMARSBOT I2C Sensor Controller Library Documentation |

Table of Contents

[1.0 LIBRARY OVERVIEW 1](#_Toc68369886)

[2.0 INSTALLATION 1](#_Toc68369887)

[3.0 FUNCTION DESCRIPTIONS 2](#_Toc68369888)

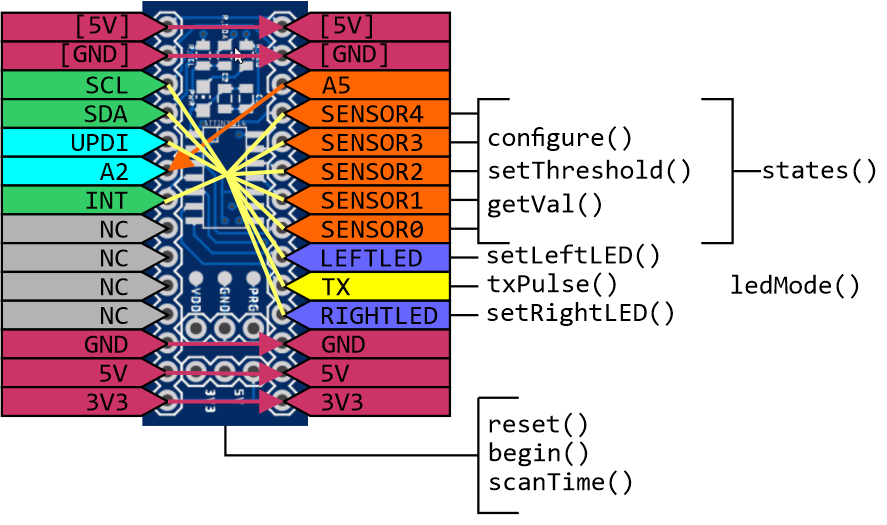
[4.0 EXAMPLES 16](#_Toc68369889)

[5.0 Application Note – Custom Sensor Board Configuration: 19](#_Toc68369890)

The ISC has a supporting Arduino library to simplify the operation. It is hoped that this documentation describes enough detail that the ISC can be incorporated in a project. However, for more comprehensive detail, please refer to the ISC Datasheet.

# LIBRARY OVERVIEW

The below diagram indicates schematically how the functions within the Arduino ISC Library may be used in conjunction with an ISC board.



# INSTALLATION

The ISC library may be installed in the same way as any other Arduino Library. Download and unzip ISC folder, copy the folder into the ‘lib’ folder within the Arduino installation. Once the Arduino software is restarted, the library will be available to include and an Examples folder is available with some example code with comments explaining functionality.

To use this library, include the following at the start of the code:



# FUNCTION DESCRIPTIONS

## reset()

Description

This function forces the ISC to a reset state, allowing the initial setup to be performed.

Syntax

void reset(uint8\_t address);

Parameters

address 7-bit I2C address for ISC device – printed on device.

Returns

N/A

Example

Isc.reset(0x50); //Reset ISC with I2C (7-bit) Address of 0x50

## begin()

Description

This function starts the ISC in run mode, confirming that setup is complete. It defines the sensor board that is attached.

Syntax

void begin(uint8\_t boardType);

Parameters

boardType Type of sensor board connected to the ISC:

basicLineSensor UKMARS Basic Line Sensor Board

basicWallSensor UKMARS Basic Wall follower Board

spLineSensor Line Sensor board by S. Pithouse

customBoard Undefined board

Returns

N/A

Example

Isc.begin(basicLineSensor); //Start with Basic Line sensor attached

## scanTime()

Description

This function returns the most recent time taken for the ISC to complete a full cycle. The units are µs.

Syntax

int scanTime();

Parameters

N/A

Returns

Most recent time taken for the ISC to complete a full cycle (µs).

Example

int myTime = Isc.scanTime(); //Store most recent scan time in myTime

## setThreshold()

Description

This function allows the sensor threshold to be set for a specific sensor.

Syntax

void setThreshold(uint8\_t sens, uint16\_t threshVal);

Parameters

Sensor

SENSOR0 See diagram in Section ?? – A0 on Sensor Board

SENSOR1 See diagram in Section ?? – A0 on Sensor Board

SENSOR2 See diagram in Section ?? – A0 on Sensor Board

SENSOR3 See diagram in Section ?? – A0 on Sensor Board

SENSOR4 See diagram in Section ?? – A0 on Sensor Board

threshVal

Value between 0 – 1024

Returns

N/A

Example

Isc.setThreshold(SENSOR0, 500);

## getVal()

Description

This function returns the most recent sensor value for the specified sensor.

Syntax

int getVal(uint8\_t sensor);

Parameters

Sensor

SENSOR0 See diagram in Section ?? – ‘A0’ on Sensor Board

SENSOR1 See diagram in Section ?? – ‘A1’ on Sensor Board

SENSOR2 See diagram in Section ?? – ‘A2’ on Sensor Board

SENSOR3 See diagram in Section ?? – ‘A3’ on Sensor Board

SENSOR4 See diagram in Section ?? – ‘A4’ on Sensor Board

Returns

Value of specified sensor

Example

leftSensor = Isc.getVal(SENSOR0);

## ledMode()

Description

This function controls the Indicator LED function

Syntax

void ledMode(uint8\_t mode);

Parameters

mode

MASTERCTRL MCU has control

FREQOUT Scan frequency output on LEDs

LEDOFF Turn LEDs off

LEDBRDCTRL Allow board specific LED control

FASTBLINK Start LEDS blinking fast

SLOWBLINK Start LEDs blinking slowly

Returns

N/A

Example

Isc.ledMode(FASTBLINK);

## setLeftLED()

Description

This function controls the Indicator LED function. Reads current led state, sets into Master control mode and turns Left LED on.

Syntax

void setLeftLED(uint8\_t state);

Parameters

state

HIGH Turn LED on

LOW Turn LED off

Returns

N/A

Example

Isc.setLeftLED(HIGH);

## setRightLED()

Description

This function controls the Indicator LED function. Reads current led state, sets into Master control mode and turns Left LED on.

Syntax

void setRightLED(uint8\_t state);

Parameters

state

HIGH Turn LED on

LOW Turn LED off

Returns

N/A

Example

Isc.setRightLED(HIGH);

## configure()

Description

This function allows individual sensor setup to be configured. There are keywords that may be combined with ‘+’ as shown.

Syntax

Isc.configure(uint8\_t sensor, uint8\_t config);

Parameters

sensor

SENSOR0 See diagram in Section ?? – ‘A0’ on Sensor Board

SENSOR1 See diagram in Section ?? – ‘A1’ on Sensor Board

SENSOR2 See diagram in Section ?? – ‘A2’ on Sensor Board

SENSOR3 See diagram in Section ?? – ‘A3’ on Sensor Board

SENSOR4 See diagram in Section ?? – ‘A4’ on Sensor Board

config

enb Enable sensor – ISC will read the value

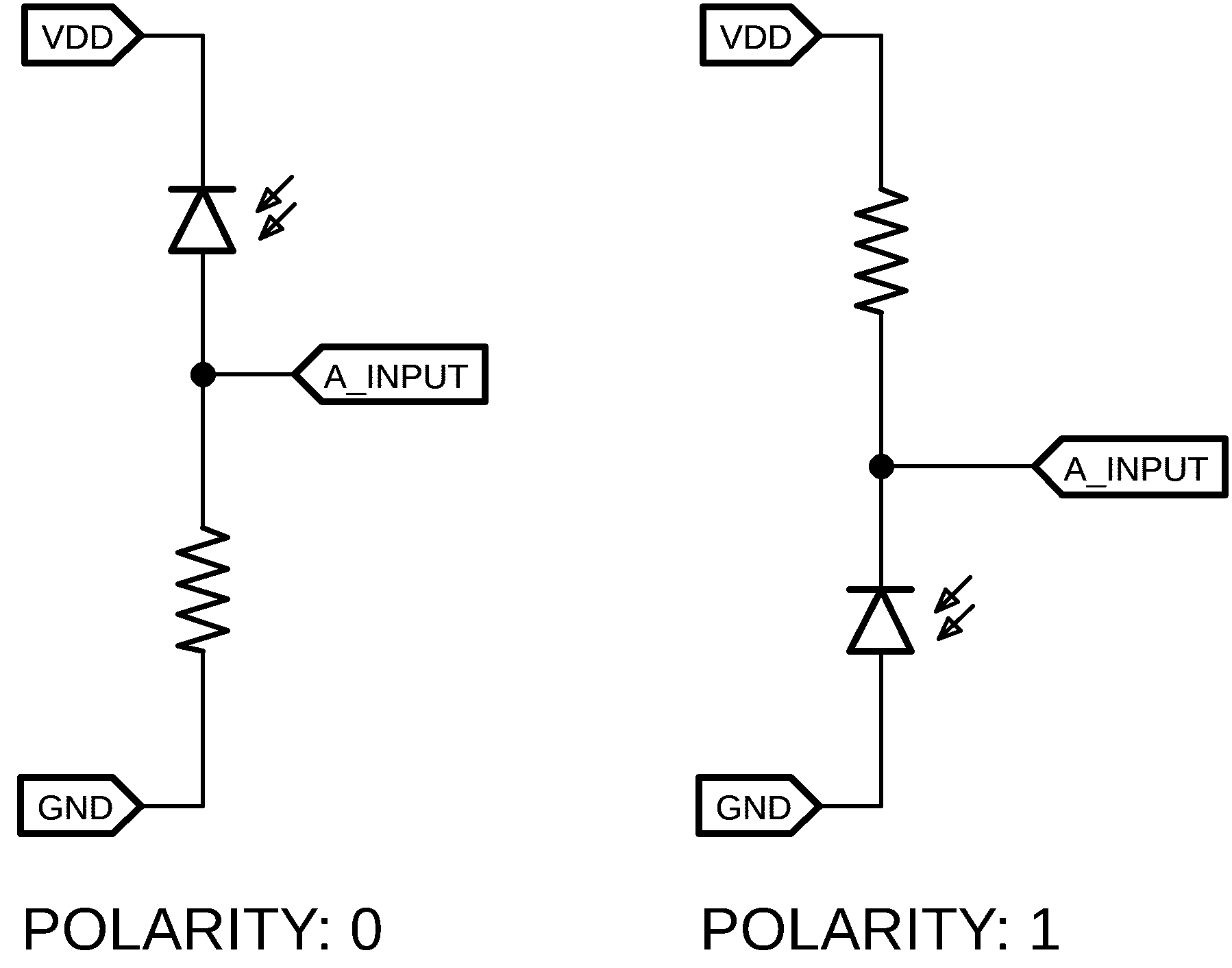
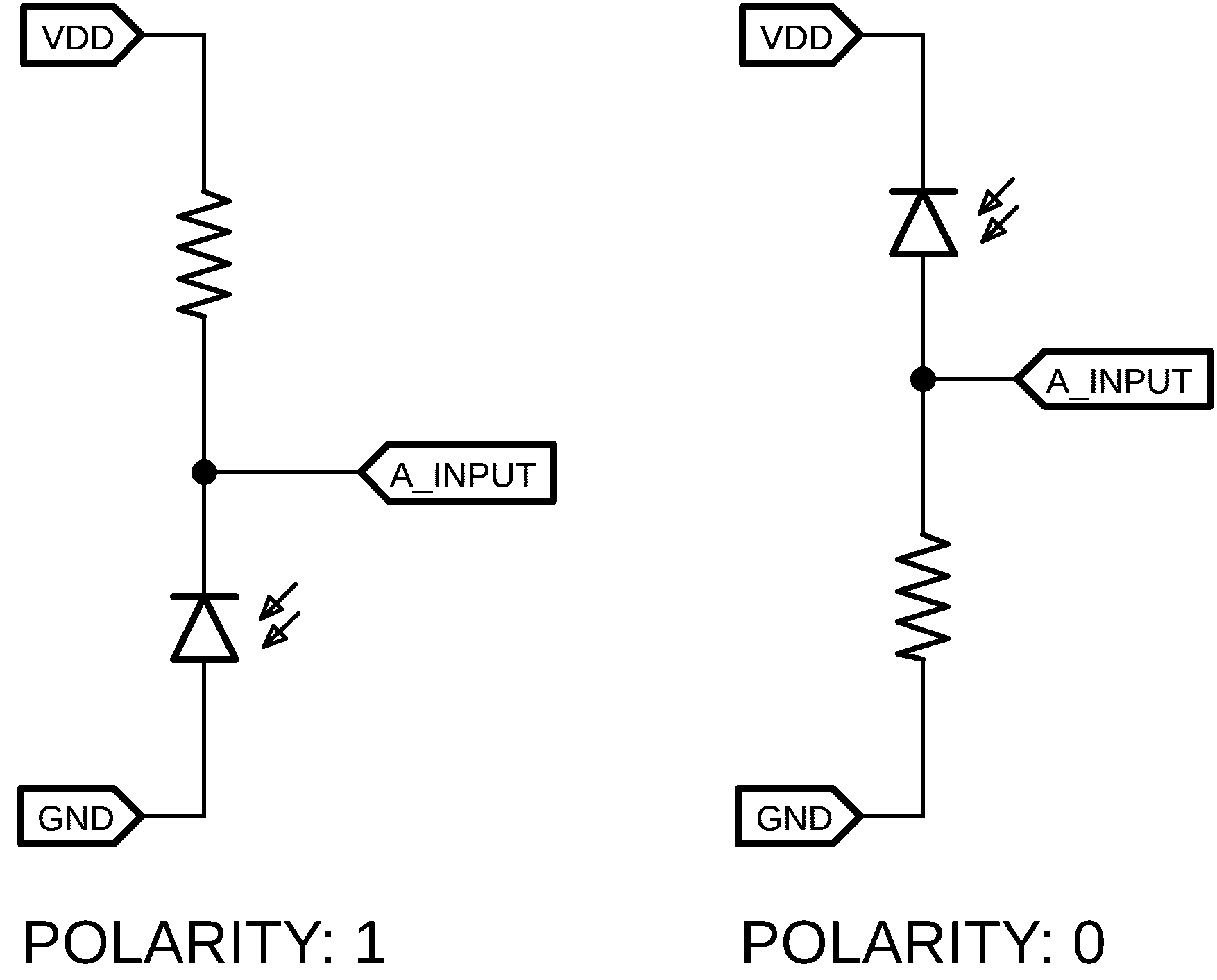
lowRes Reduce value to 8-bit

txEnb Enable Transmitter to help remove ambient light

fallingInterrupt Sensor will cause interrupt on 1->0 state change

risingInterrupt Sensor will cause interrupt on 0->1 state change

flip [See Polarity]

 ??? or ??? 

Returns

N/A

Example

Isc.configure(SENSOR0, enb + txEnb + risingInterrupt); //Marker Sensor

## txPulse()

Description

This function allows the time in µs that the Tx emitter is on before the first sample is taken to be adjusted.

Syntax

void txPulse(uint8\_t length);

Parameters

Length

Amount of time in us…??

Returns

N/A

Example

Isc.txPulse(100);

## states()

Description

This function gets the current states of the sensors and returns a byte with 1 representing HIGH sensor state.

Syntax

uint8\_t states();

Parameters

N/A

Returns

Byte with bits indicating current sensor state

Example

leftState = Isc.states() & 0b1;

## read()

Description

Reads a byte (or pair of bytes) from a specified register within the ISC.

Syntax

int read(uint8\_t regAddr, uint16\_t numVals);

Parameters

regAddr

Register address to read from

numVals

May be 1 or 2, for the number of consecutive registers to read (ie. 2 when a 2-byte

value is to be read).

Returns

The value stored in that location (int)

Example

int timeTaken = Isc.read(SCANTIME, 1);//Read 1 byte at address SCANTIME

## write()

Description

Writes a byte (or pair of bytes) to a specified register within the ISC.

Syntax

void write(uint8\_t regAddr, uint16\_t data, uint8\_t numVals);

Parameters

regAddr

Register address to write to

data

Data to write

numVals

May be 1 or 2, for the number of consecutive registers to write (ie. 2 when a 2-byte

value is to be written).

Returns

N/A

Example

Isc.write(SENS0THRSH, 500, 2);//Write 500 (2 bytes) for Sensor 0 Threshold

# EXAMPLES

## Turn Left LED on

Initialise the ISC and set Left LED on.

#include <ISC.h>

ISC Isc;

void setup(){

Isc.reset(0x50); //Establish communication and Reset ISC (Address 0x50)

Isc.begin(basicLineSensor);

setLeftLED(HIGH);

}

void loop(){

}

## Flash LEDs slowly

Initialise the ISC and set LED mode to SLOWBLINK.

#include <ISC.h>

ISC Isc;

void setup(){

Isc.reset(0x50); //Establish communication and Reset ISC (Address 0x50)

Isc.begin(basicLineSensor);

ledMode(SLOWBLINK);

}

void loop(){

}

## Single Sensor Read

Initialise the ISC and read sensor 0 value every 100ms.

#include <ISC.h>

ISC Isc;

void setup(){

Serial.begin(115200); //Start communications over Serial Port

Isc.reset(0x50); //Establish communication and Reset ISC (Address 0x50)

Isc.setup(SENSOR0, enb+txEnb); //Setup Left Sensor, enabled with Tx enabled

Isc.txPulse(10); //10µs Tx on time before sampling begins

Isc.begin(basicLineSensor);

}

void loop(){

Serial.println(Isc.getVal(SENSOR0));

delay(100);

}

## Obtain ISC scan time

Initialise the ISC and report the scan time every 100ms

#include <ISC.h>

ISC Isc;

void setup(){

Serial.begin(115200); //Start communications over Serial Port

Isc.reset(0x50); //Establish communication and Reset ISC (Address 0x50)

Isc.begin(basicLineSensor);

}

void loop(){

Serial.println(Isc.scanTime(SENSOR0));

delay(100);

}

# TEXT – KEYWORDS

Isc.configure(SENSOR0, enb + txEnb + risingInterrupt); //Configure Marker Sensor

# FUNCTIONS

**editor.keyword1.style = #d35400,bold**

editor.data\_type.style = #d35400,bold

# METHODS

editor.keyword2.style = #D35400,plain

editor.function.style = #d35400,plain

# STRUCTURES

editor.keyword3.style = #5E6D03,plain

editor.reserved\_word.style = #5E6D03,plain

# TEXT - LITERALS

editor.literal1.style = #00979C,plain

editor.literal2.style = #00979C,plain

editor.variable.style = #00979C,plain

editor.reserved\_word\_2.style = #00979C,plain

editor.literal\_boolean.style = #00979C,plain

editor.literal\_char.style = #00979C,plain

editor.literal\_string\_double\_quote.style = #005C5F,plain

editor.preprocessor.style = #5E6D03,plain

# http://www.arduino.cc/

editor.url.style = #0000ff,underlined

# e.g. + - = /

editor.operator.style = #434f54,plain

# ?? maybe this is for words followed by a colon

# like in case statements or goto

**editor.label.style = #7e7e7e,bold**

# TEXT - COMMENTS

editor.comment1.style = #434F54,plain dk grey /\*\*/

editor.comment2.style = #95a5a6,plain light grey //

# LINE STATUS - editor line number status bar at the bottom of the screen

linestatus.font = SansSerif,plain,10

linestatus.height = 20

# Application Note – Custom Sensor Board Configuration:

