am

Eval Kit Manual

AS5600

Adapter Board

AS5600-POTUINO

Table of Contents

Revision History

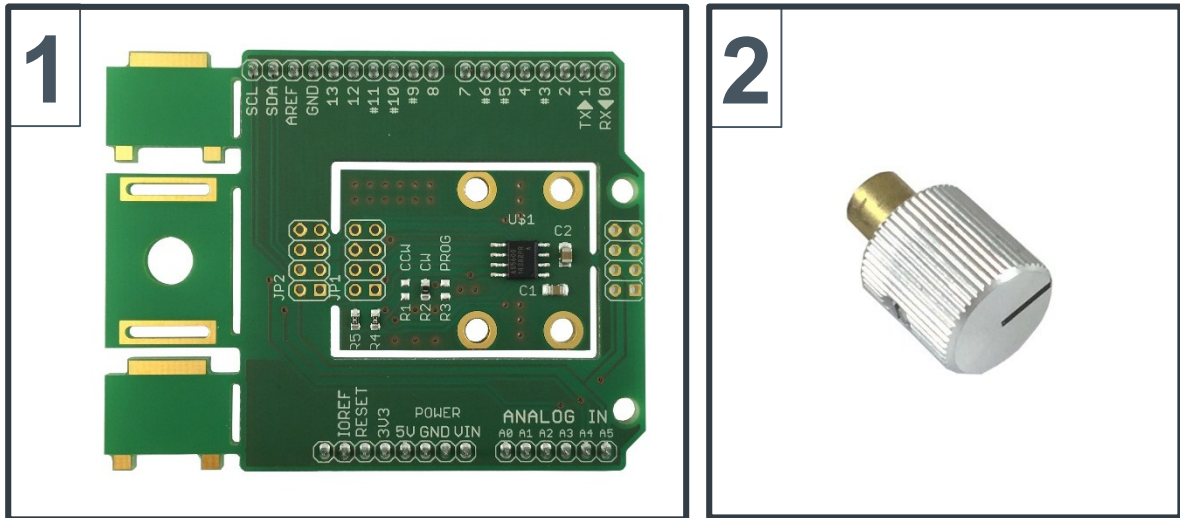| Revision	Date	Owner	Description
1.0	18.03.2015	mzie	Initial version

1 Introduction

The POTUINO is a potentiometer shield compatible with the Arduino UNO board. It was designed to evaluate the AS5600 Smart Potentiometer IC.

1.1 Kit Content

Figure 1: Kit content

Pos.	Item	Description
1	AS5600-POTUINO	Arduino shield
2	RMH05-DK	Rotary magnet holder

Note: An Arduino UNO board (not included) is required to use the POTUINO shield.

2   Board description

The POTUINO shield allows evaluation of the AS5600 position sensor. This Arduino shield is fully assembled with the AS5600 IC and its necessary external components.

The Arduino shield includes different breakout options. A breakout magnet holder is available on the left side of the PCB and in addition, an adapter board breakout is possible.

The headers on the bottom side of the PCB are used for mounting as well as for electrical connection to the Arduino UNO.

Figure 2: AS5600-POTUINO shield

Breakout magnet holder

AS5600 sensor IC
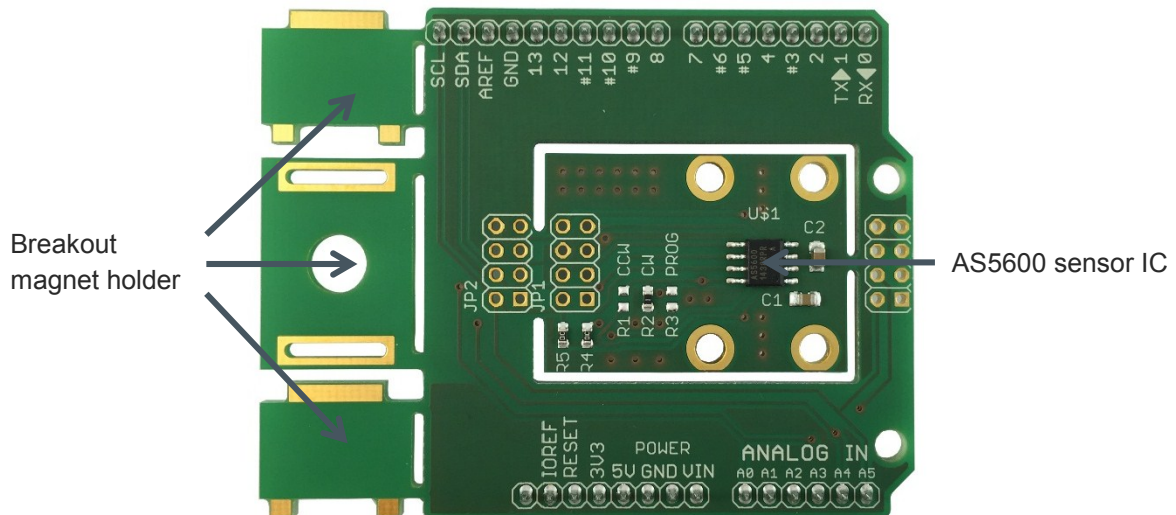
3   Software

Custom firmware source code can be generated using the Arduino IDE. An example code and Arduino library is available for reference. In addition, the AS5600 LabVIEW Evaluation GUI software can be used to readout and configure the AS5600.

3.1   LabVIEW

The AS5600 LabVIEW Evaluation GUI supports the Arduino UNO. The latest version of the software can be downloaded from the ams webpage.

Before the Arduino UNO can be used together with the POTUINO shield the LabVIEW interface firmware has to be flashed onto the Arduino. Easiest way to do this is to download the free software tool **XLoader**.

Figure 3: XLoader for flashing the Arduino UNO
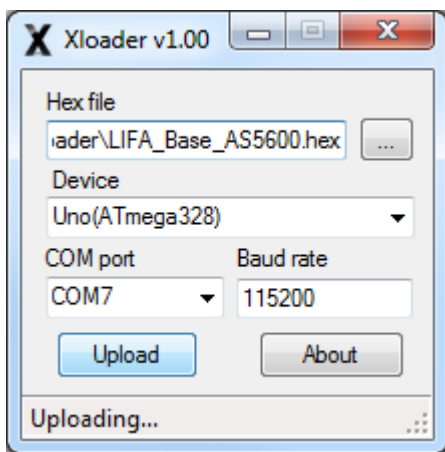
Choose the **LIFA_Base_AS5600.hex** file which is also available for download from the ams webpage. Then choose the **Uno(ATmega328)** in the Device dropdown menu. After this choose the correct **COM port** which can easily be checked in the Windows Device Manager and finally click **Upload**.

After this procedure the Arduino UNO is ready for usage with the LabVIEW Evaluation GUI of the AS5600.

If the Hardware Settings window opens, the switch has to be moved to "Arduino UNOr3". This is show in Figure 4: AS5600 Evaluation GUI – Hardware Settings below.

A green tick indicates successful communication with the sensor. For further information about the AS5600 LabVIEW Evaluation GUI please refer to the User Manual.

Figure 4: AS5600 Evaluation GUI – Hardware Settings
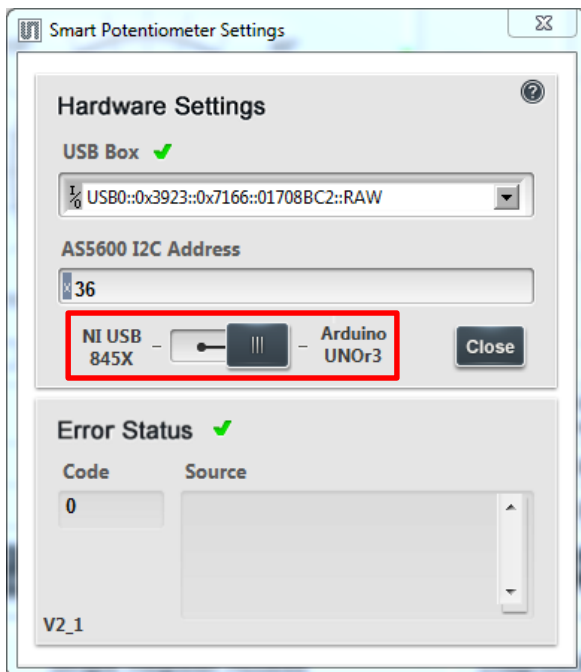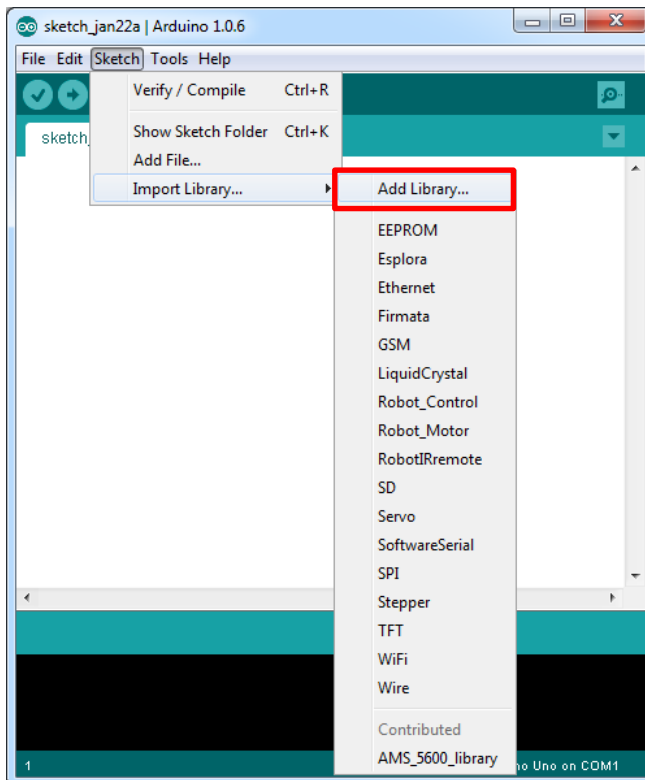
3.2   Using the Arduino IDE

3.2.1 Installing the Arduino library

The Arduino library for the AS5600 sensor can be downloaded from the ams webpage.

To add an existing library open the Arduino IDE, click **Sketch** in the menu bar. Then **Import Library** and **Add Library**.

Then choose the **AMS_5600_library.zip** and open it.

Figure 5: Adding the AS5600 library

When including the header file to a new sketch all the functions of the library are available.

e.g.

```
#include <AMS_5600.h>
```

3.2.2 Reading out the AS5600 sensor

Source code examples are also available for download on the ams webpage.

Following source code shows an easy example of reading out the sensor:

```
/*-----------------------------------------------------
FILE:  AS5600

Author: Mark A. Hoferitza, Field Application Engineer, ams AG
www.ams.com
Date: 27 May 2014

Description:  Development of sketches for AS5600 "Potuino"

Read Raw Angle and Angle.
Single Value, no averaging.
-----------------------------------------------------*/

#include <Wire.h>

int AS5600 ADR = 0x36;
const int raw_ang_hi = 0x0c;
const int raw_ang_lo = 0x0d;
const int ang_hi = 0x0e;
const int ang_lo = 0x0f;
const int stat = 0x0b;
const int agc = 0x1a;
const int mag_hi = 0x1b;
const int mag_lo = 0x1c;

void setup(){
 Serial.begin(9600);
 Wire.begin();
 }

void startup(){
}

void loop(){

//  Wire.beginTransmission(AS5600_ADR);
//  Wire.write(0x);
//  Wire.write(0x00);
//  Wire.endTransmission();


//**********************************************************
// Read Raw Angle Low Byte
  Wire.beginTransmission(AS5600_ADR);
  Wire.write(raw_ang_lo);
  Wire.endTransmission();
  Wire.requestFrom(AS5600_ADR, 1);
  while(Wire.available() == 0);
  int lo_raw = Wire.read();

// Read Raw Angle High Byte
  Wire.beginTransmission(AS5600_ADR);
  Wire.write(raw_ang_hi);
  Wire.endTransmission();
  Wire.requestFrom(AS5600 ADR, 1);
  while(Wire.available() == 0);
  word hi_raw = Wire.read();

  hi_raw = hi_raw << 8;  //shift raw angle hi 8 left
  hi_raw = hi_raw | lo_raw;  //AND high and low raw angle value

//**********************************************************
// Read Angle Low Byte
```

```
  Wire.beginTransmission(AS5600_ADR);
  Wire.write(ang lo);
  Wire.endTransmission();
  Wire.requestFrom(AS5600_ADR, 1);
  while(Wire.available() == 0);
  int lo_ang = Wire.read();

//Read Angle High Byte
  Wire.beginTransmission(AS5600_ADR);
  Wire.write(ang_hi);
  Wire.endTransmission();
  Wire.requestFrom(AS5600 ADR, 1);
  while(Wire.available() == 0);
  word hi_ang = Wire.read();

  hi_ang = hi_ang << 8;
  hi_ang = hi_ang | lo_ang;

//***********************************************
//Read Magnitude High Byte
  Wire.beginTransmission(AS5600_ADR);
  Wire.write(mag_hi);
  Wire.endTransmission();
  Wire.requestFrom(AS5600 ADR, 1);
  while(Wire.available() == 0);
  word hi_mag = Wire.read();

//Read Magnitude Low Byte
  Wire.beginTransmission(AS5600_ADR);
  Wire.write(mag lo);
  Wire.endTransmission();
  Wire.requestFrom(AS5600_ADR, 1);
  while(Wire.available() == 0);
  word lo_mag = Wire.read();

  hi mag = hi mag << 8;
  hi_mag = hi_mag | lo_mag;

//***********************************************
//Read AGC
  Wire.beginTransmission(AS5600_ADR);
  Wire.write(agc);
  Wire.endTransmission();
  Wire.requestFrom(AS5600 ADR, 1);
  while(Wire.available() == 0);
  int agc_val = Wire.read();

//***********************************************
// Read Status Bits
  Wire.beginTransmission(AS5600_ADR);
  Wire.write(stat);
  Wire.endTransmission();
  Wire.requestFrom(AS5600_ADR, 1);
  while(Wire.available() == 0);
  int stat_val = Wire.read();

  stat_val = stat_val & 0xa0;  //Mask off all unnecessary bits

//Print Values Received
 float Raw_Ang = hi_raw *0.0879;
 float Ang = hi_ang * 0.0879;

  Serial.print("  AS5600 Raw Angle = ");   Serial.println(Raw Ang, DEC);
  Serial.print("      AS5600 Angle = ");   Serial.println(Ang, DEC);
  Serial.print("  AS5600 Magnitude = ");  Serial.println(hi_mag, HEX);
  Serial.print("        AS5600 AGC = ");  Serial.println(agc_val, HEX);
  Serial.print("AS5600 Status Bits = ");  Serial.println(stat_val, BIN);
  Serial.println();
  Serial.println();

  delay(2000);
}
```
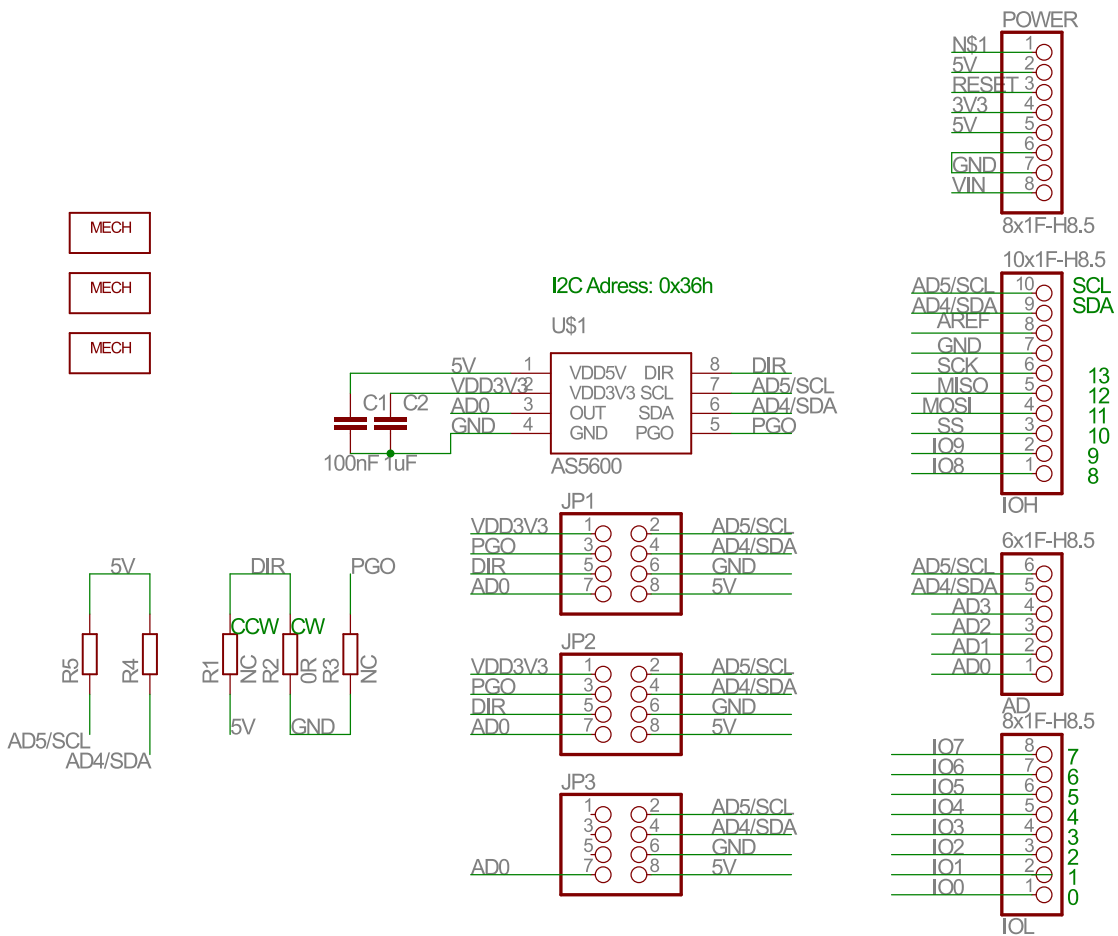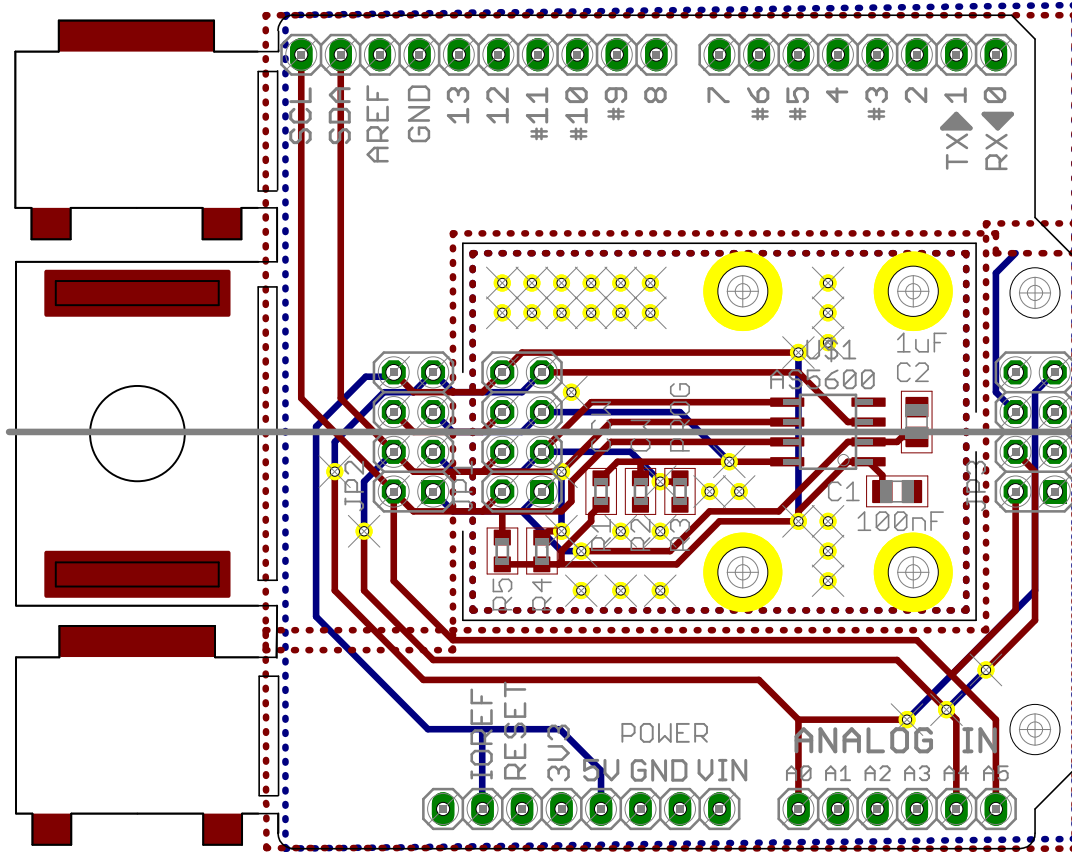
4 AS5600-POTUINO Hardware

4.1 AS5600-POTUINO schematics

Figure 6: AS5600-POTUINO schematics

4.2   AS5600-POTUINO PCB layout

Figure 7: AS5600-POTUINO PCB layout

5   Ordering & Contact Information

Ordering Code	Description
AS5600-POTUINO	AS5600 Eval Kit Adapter Board

Buy our products or get free samples online at:

www.ams.com/ICdirect

Technical Support is available at:

www.ams.com/Technical-Support

Provide feedback about this document at:

www.ams.com/Document-Feedback

For further information and requests, e-mail us at:

ams_sales@ams.com

For sales offices, distributors and representatives, please visit:

www.ams.com/contact

Headquarters

ams AG

Tobelbaderstrasse 30

8141 Unterpremstaetten

Austria, Europe

Tel: +43 (0) 3136 500 0

Website: www.ams.com

6 Copyrights & Disclaimer

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[ams OSRAM](#):
  [AS5600-SO_POTUINO](#)