Waveshare Accessory Shield unofficial Documentation

Version 1.00

Goal

Having recently purchased the Waveshare Accessory Shield ("AS") for a personal project, I found the online documentation for the shield - both Hardware & Software - to be very sparse and confusing (maybe because this product was discontinued.).

Since I consider the AS a fantastic starting point for Arduino beginners, I've decided to create this alternative unofficial documentation for the Shield in the hope that this will benefit the community.

For this project, the AS is mounted on an Arduino Uno R3 board..

For questions and info, contact me at ran@ranlevi.com, or follow me on Twitter: @ranlevi.
Ran.

Board content

The AS has 3 connectors, compatible with the Uno R3 board.

Communication between the Uno's ATmega3289P microcontroller and the various devices on the AS is via I2C (note: the official docs say SPI, but as far as I can tell that's not the case.)

The board implements the following features (see Fig. 1 for location on the board):

- 5-way joystick: Up, Down, Right, Left, Press. (9)
- Adjustable potentiometer.(6)
- A sound-producing buzzer. (5)
- 3 color LED (RGB). (4)
- Temperature Sensor. (8)
- 3-axis accelerometer. (7)
- Real Time Clock. (15)
- 0.96 inch, 128x64 Yellow/Blue OLED screen. (3)
- XBee interface. (2)

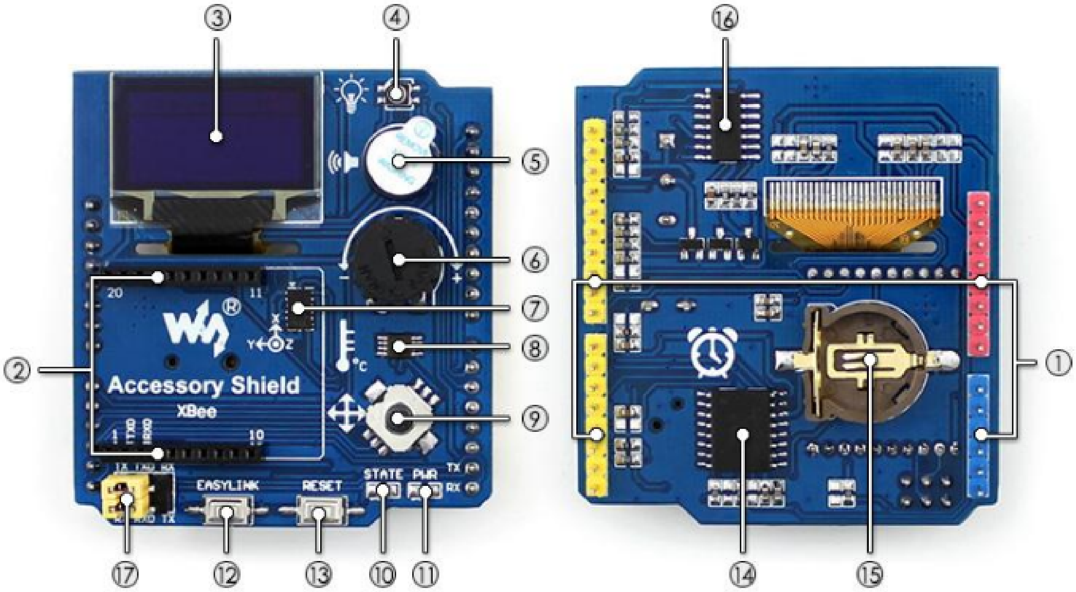

Fig. 1: Location of various features on the AS (Source: Waveshare Accessory Shield User Manual)

Features in-depth

5-Way Joystick

Hardware

The joystick is connected, via AS H3 connector, to the Uno’s following Analog pins:

- A1
- A2
- A3
- A4
- A5

Pins are initialized as HIGH. A press will change the relevant pin’s value to LOW.

Note: A4 & A5 also serve as I2C communication, and are also routed to the AS H2 connector, pins 9 & 10 respectfully.

Software

The joystick feature does not require any additional library.

Demo code: AS_joystick_demo.ino

Adjustable Potentiometer

Hardware

An on-board 10k adjustable resistor, connected (according to the schematics) to a 3.3V supply, and the A0 analog pin.

All of the Uno’s analog pins are connected to a 10-bit Analog-to-Digital converter, which converts the voltage value detected on the pin to one of 2^10 (=1024) digital values. When the Uno is powered from the USB port (5V), the converter’s resolution is 4.88mV. This means that the digital values received from the converter will be in the range of [0, 3300mv/4.88mv = 676].

Software

No additional library required.

Demo code: AS_adjustable_potentiometer_demo.ino

RGB LED

Hardware

The RGB LED is driven by a P9813 driver, which communicates with the Uno’s uController via a serial interface on digital pins D5 (data) & D6 (clock). It can modify the LED’s hue, saturation and brightness.

The power to the LED is enabled by digital pin D12.

Software

An additional library, ChainableLED, is required.

Demo code: AS_RGB_Demo.ino

Buzzer

Hardware

The buzzer is connected to Digital pin D11, which is driven as a PWM with various frequencies.

Software

No additional library required.

Demo code: AS_Buzzer_Demo.ino

Temperature Sensor

Hardware

LM75 is a chip-mounted temperature sensor, which communicates with the uController via I2C comm., via Analog pins A5 & A6.
The LM75 includes an 11-bit Analog-to-Digital converter (ADC), which allows a temperature resolution of 0.125 deg. Celsius).

The read temperature is displayed via the Serial Monitor. To change the temp, touch the LM75 (note the thermometer icon next to it on the PCB.)

Software

Two additional libraries required:

- Wire.h (bundled with the Arduino by default),
- Lm75.h

Demo code: AS_Temp_Sensor_Demo.ino

OLED Display

Hardware

The OLED display is (probably) an Adafruit SSD1306 OLED display.
Communication with the uController is via I2C, using pins A5 & A6.
Digital pin D7's function is unclear.
Digital pin D8 acts as adress 0 (SA0) for the I2C slave adress selection.

Software

This feature requires the following libraries:

- Wire.h (I2C library),
- Adafruit_GFX.h (basic graphics functions, common to all Adafruit displays.)
- Adafruit_SSD1306.h (Device driver)

Demo code: AS_Temp_Sensor_Demo.ino

XBee Interface

This feature is not tested at this time.

Real-Time Clock

This feature is not tested at this time.