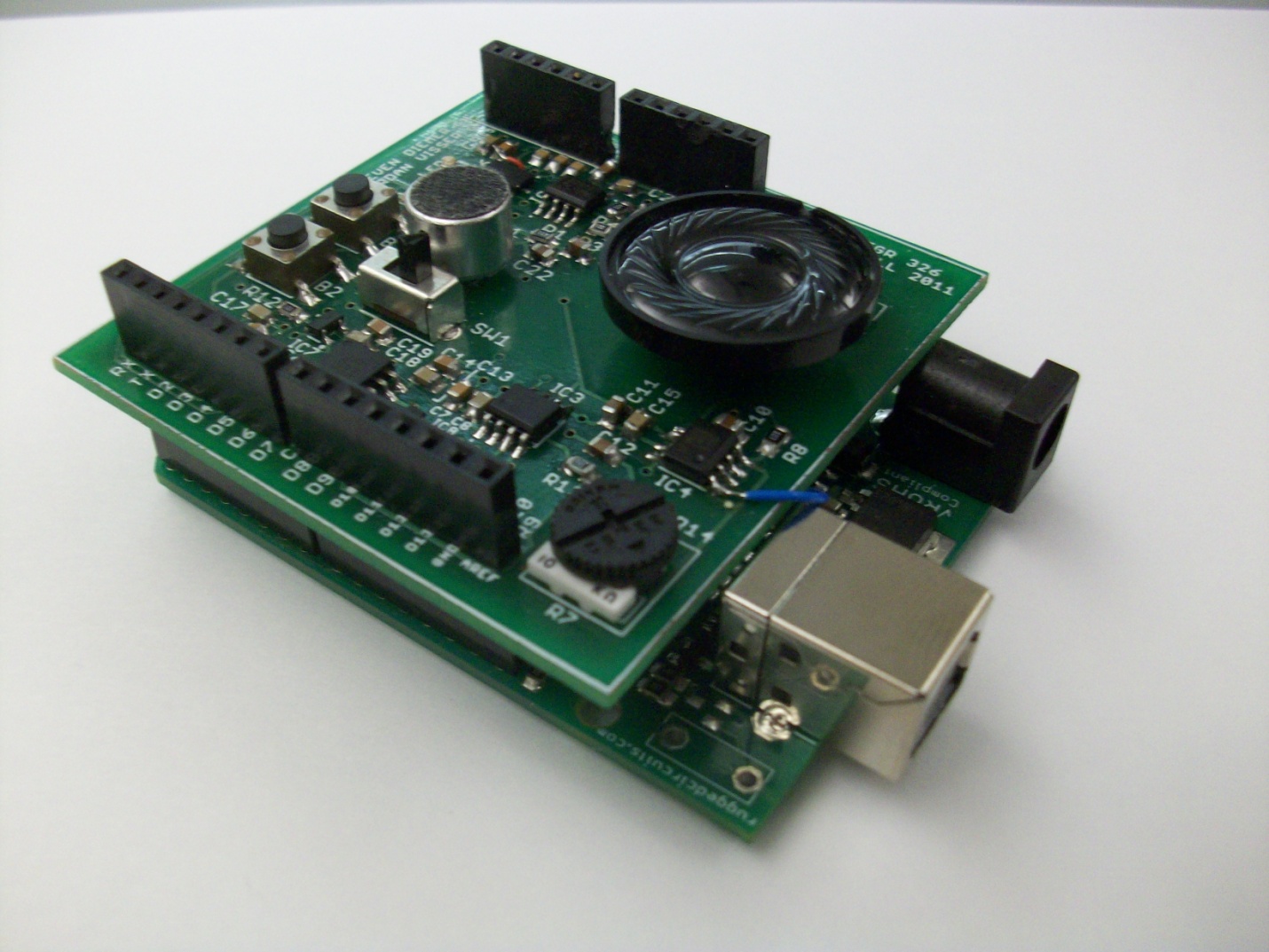
***Wave-Motion Shield***

***User’s Manual***

******

Revision 1.0

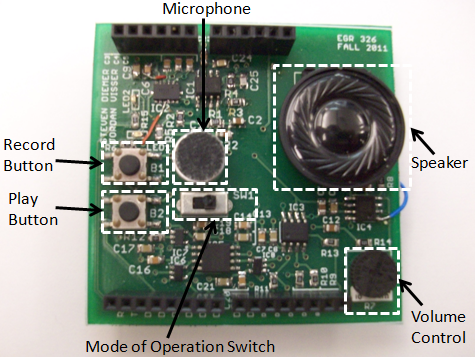
Dec. 12, 2011

**1.0 Introduction**

Thank you for purchasing the Wave-Motion Arduino Shield! Your shield is 100% assembled and has been quality control tested. Please take precautions as not to touch any components on the board other than the 2-position switch and the record and play buttons. Also, the shield should be kept in an ESD protection bag whenever not in use.

**2.0 Getting Started**

The shield comes pre-loaded with the standard Wave-Motion software. As such, the shield is ready to be mounted to an Arduino Uno and powered via the USB interface on the Uno. Before powering the shield, familiarize yourself with some of the basic hardware on the Wave-Motion. Figure 1 shows the location of the main hardware components you will be interfacing.



**Figure 1: Basic Hardware Identification**

**2.1 Recording a Message**

To record a message, ensure that the Mode of Operation Switch is in the leftmost position. This position places the shield in Record Mode. Next, hold the Record Button to record a message. The button must be pressed for the entirety of the message. A red LED will turn on indicating you are, in fact, recording. You are able to record for up to 60 seconds.

**2.2 Playing back a Message**

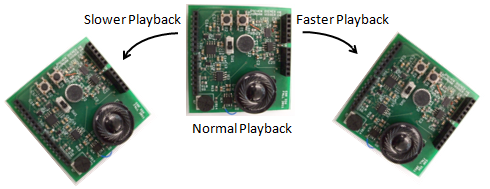
Once you have finished a recording, the message can be played back by pressing the Play Button. The entire recording is then played back from start to finish. Volume can be increased or decreased by turning the Volume Control knob. Turning the knob clockwise increases volume while turning the knob counterclockwise decreases the volume of the playback.

**3.0 Controlling Playback Speed**

The Wave-Motion has two modes in which playback speed can be controlled. These two modes are *tilt-control mode* and *circle-wacky-winding mode.* When the shield is first powered up, the shield will be in *tilt-control mode.* To toggle modes, hold the shield upside-down for 3 seconds. A laser beam sound will then play indicating you have changed modes.

**3.1 Tilt-Control Mode**

To change the playback speed of a message in *tilt-control mode,* simply hold the device upright (the USB port of the Uno should be pointed straight down) to play back the message at its native recording speed. To playback the message faster, tilt the shield to the right. To playback the message slower, tilt the shield to the left. This mode of playback is summarized by Figure 2.



**Figure 2: Tilt-Control Playback Mode Operation**

**3.2 Circle-Wacky-Winding Mode**

To change the playback speed of a message in *circle-wacky-winding mode*, simply hold the device flat (the speaker should be pointing toward the ceiling) and begin making a circular motion with the shield. You will notice that the faster you wind the shield in a circle, the faster the message is played back.

**4.0 Gesture Recognition Sound Effect Mode**

By placing the Mode of Operation Switch in its rightmost position, the shield will be put into Gesture Recognition Sound Effect Mode. In this mode, the Wave-Motion can be moved in certain ways to playback unique sound effects.

**4.1 Laser-Pistol Sound Effect**

The laser-pistol sound effect can be triggered by “flicking” the Wave-Motion forward. You may notice a “clicking” sound after shooting the laser-pistol three times. Don’t panic! You are simply out of ammo and need to reload. To reload, flick the shield backward (towards your face if you are holding it with the speaker told the ceiling) and you will be blessed with three more shots.

**4.2 Disc Jockey Sound Effect**

The disc jockey sound effect is triggered by moving the shield quickly left or right while holding the shield with the speaker facing the ceiling. You will notice that each direction invokes a different disc-scratching noise.

**5.0 Modifying the Wave-Motion Source Code (Advanced Users)**

As the Wave-Motion is open source, the source code is made readily available for you to download [here](https://github.com/stevendiemer/Project-326/tree/master/Code). To view/modify the source code, you will also need to download the Arduino IDE. This can be found at <www.arduino.cc>. Additional supporting documents for the Wave-Motion can also be found [here](http://www.egr.gvsu.edu/~diemerst/wordpress/?page_id=16) (the Wave-Motion WordPress site).

**6.0 Uploading Sound Effects to the Wave-Motion Shield (Advanced Users)**

New sound effects can be added to the Wave-Motion shield by storing new sound clips to the flash memory. In order to upload new sound clips, the files must be in .wav (uncompressed) format and must also be less than 32kB in size.

**6.1 Breaking a .wav File into Byte Values**

Once a sound clip is obtained, it must be broken into individual byte values that are offset to range between 0 and 255 (standard unsigned 8-bit values). This can be done by using MATLAB or other software to break the files into bytes (You can download a voicebox folder containing a readwav.m file that will do this for you). These bytes can be placed directly into the PROGMEM function seen in the SoundEffects.pde file in the Wave-Motion’s code. These numbers only need to be separated by commas. The numOfBytes variable in the code must be set to the number of bytes counted in the sound effect.

**6.2 Adjusting Memory Storage Location for Sound Effects**

In order to add new sound effects, a storage location must be computed to prevent overwriting other sound effects. The memory location is in hexadecimal format. Upon purchase of the Wave-Motion Shield the last used memory location for a sound effect is 0x2441EE. Any address after this is valid for storage of sound effects. Each storage location must be subsequently calculated for each sound effect based upon the number of bytes present in each sound clip.

**6.3 Uploading Sound Effects to Wave-Motion**

Once the byte values are placed in the PROGMEM function in the SoundEffects.pde file and the memory location has been adjusted, the bytesToMemory() function call located in Main\_1\_5.pde file (setup() function) must be uncommented and the code must simply be uploaded to the Arduino. This will safely store and protect the sound effect in memory. However, upon completion of upload, the bytesToMemory() function call must be commented out again.