267_writeup.org

N-CRITSER

<2014-05-06 Tue>

1 Abstract "TALK-A-LOT-BOT" v.0

My original goal for this project was to learn the underlying code for the Arduino platform while creating an anthropomorphic entity which "speaks" and simulates eye movement while interacting with a user. The general aspects of this I have achieved, although I have adopted the usage of higher level libraries and forgone the learning of lower level aspects of the platform.

Currently the Talk-a-lot-Bot has as its central processing unit a Rev3 Arduino Uno with an Ada-fruit Waveshield mounted to its I/O headers. The shield controls the robots "voice" and plays pre-recorded .WAV files, through a 3.5 mm stereo (TRS) plug that feeds the sound to and 8 ohm speaker. The .WAV files are stored on a Fat 16 Formatted SD card. The arduino sketch code processes the .WAV files and makes them playable by name. We use the file names and flag value settings in the code to determine which file should be played.

Also, connected to the Uno, are 2 8x8 LED matrices, which are I2C protocol driven. This means we can save a lot of I/O pins and still get a lot of variability in our eye movement. The I2C backPacks for the matrices use only Power,Ground, and 2 analog pins from the UNO (A4, A5), for clock and data. Each matrix has an address that is set in code as well as selected on the matrix backPack itself by making a solder bridge between open pads on the small backPack pcb. This addressing is explicitly controlled in the sketch code which makes the backPacks independent.

Housing the components are two sheets of ABS plastic. This material is a great substrate for electronics as well as oil paint and has a lot of fantastic properties. It's very maliable, easy to cut, and can dissipate heat fairly well. I have cut holes for the eye matrices and attached the UNO and other components via bolts. For the first iteration I have kept the button pad as a seperate unit attached only by plugged (not soldered wires).

On the button front, we have 3 simple push buttons, each with an LED that is set high while the button is HIGH and LOW when the button is pressed. This scheme helps the user know how long a button has been pressed, while also giving a visual debug of the button pin itself. The interaction between the Digital I/O pins have been a slight mystery. The UNO pins 11,12,13 are not technically used for the wave shield but if buttons are connect to pins, they act as an interrupt on the .WAV playing object. While, Pins 0,1 are free from this side effect behavior. This can be used as a way to cut short the Talk-A-lot-Bot if you feel he's talking too much. But otherwise it is a nuisance. And if an interrupt button is held down while the play flag for a .WAV file is being loaded, then the shield is unable to play the file and a few loop cycles are needed to reset the system.

Initially I wanted T.A.L.B. to play a Binary Search game with the user, but after struggling to adjust the buttons with the .WAV in a fun way, I decided to take some inspiration from an early soft bot known as PARRY. `http://en.wikipedia.org/wiki/PARRY`. PARRY was supposed to simulate a paranoid schyzophrenic and was one of the first bots to pass the Turing Test. Currently, I think T.A.L.B. doens't quite pass the Turing Test, but maybe version 1 will.

2 Components

2.1 Adafruit waveshield sd card module

`https://www.adafruit.com/products/94`

2.2 Speaker

Description for lady Ada- "Speaker - 3" diameter (77mm), 8 ohm impedence, good response between 200Hz to 10KHz (10KHz is the max frequency the Wave shield can make). The speaker is rated for 1W so if you want you can even stick a small amp between the shield and the speaker to boost up the volume"

2.3 Arduino Uno Rev3 Microcrontroller

2.4 2-8x8 i2c led matrices for simulated eye movement

```
https://www.adafruit.com/products/872
```

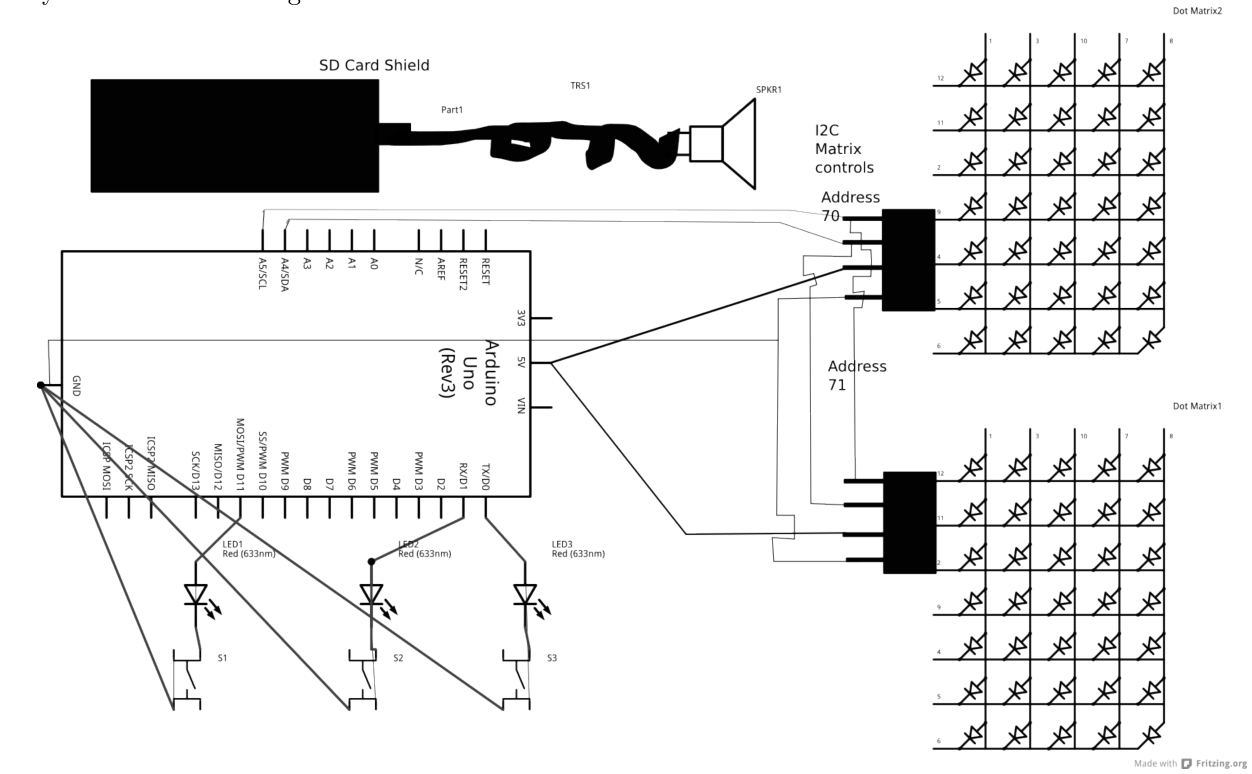
2.5 adafruit libraries for graphics and sound

3 Libraries

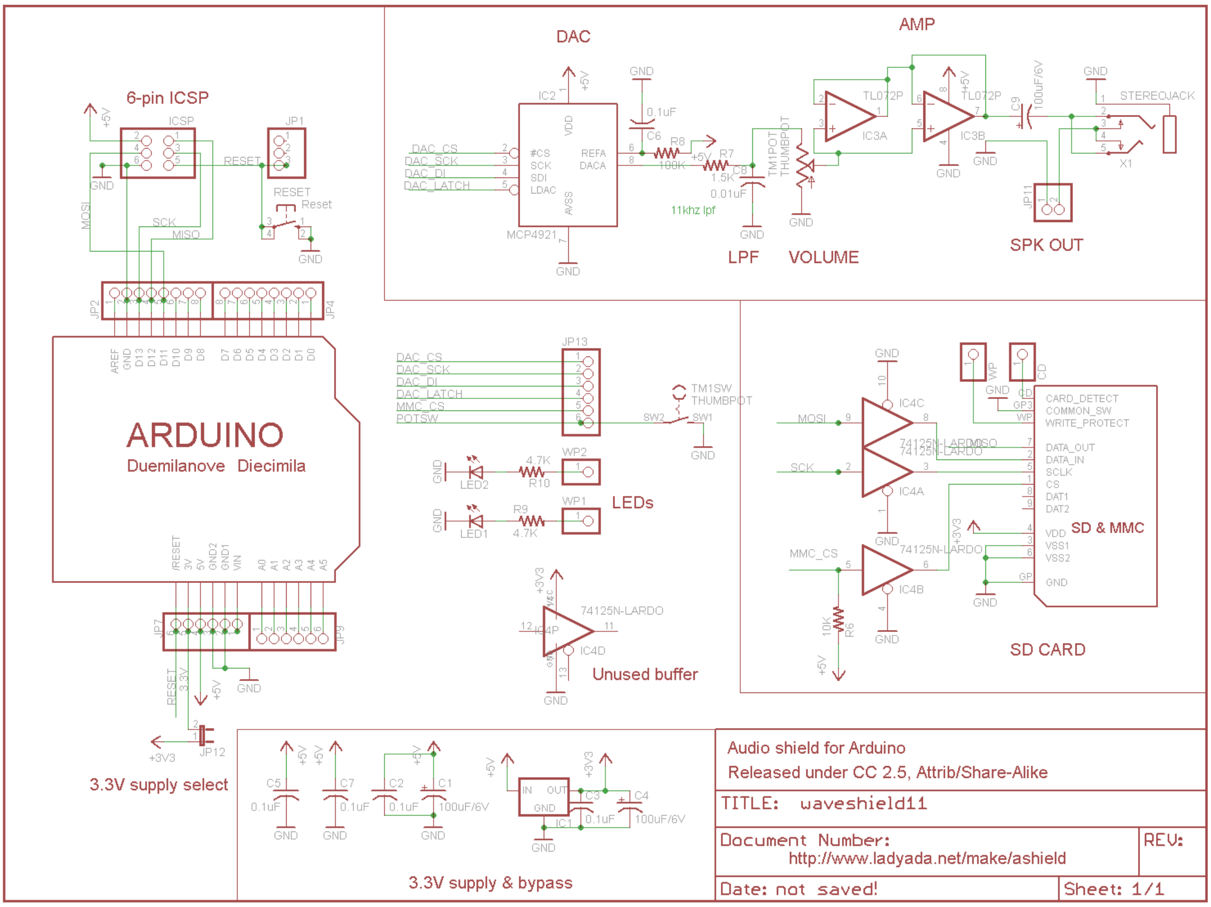
```
#include "Adafruit_LEDBackpack.h"

#include "Adafruit_GFX.h"
```

4 Wiring Diagram

The Following diagram shows only the UNO and I2C Matrices, I have included the WaveShield Schematic from Lady Ada in the following section.



5 WaveShield Diagram



6-pin ICSP
ICSP
JP1
RESET
Reset
MOSI
SCK
MISO
GND
+5V
JP2
JP4
ARDUINO
Duemilanove Diecimila
AREF GND D13 D12 D11 D10 D9 D8
D7 D6 D5 D4 D3 D2 D1 D0
/RESET 3V 5V GND2 GND1 VIN
A0 A1 A2 A3 A4 A5
JP7
JP9
RESET 3.3V +5V GND
+3V3
JP12
3.3V supply select
DAC
IC2
MCP4921
DAC_CS
DAC_SCK
DAC_DI
DAC_LATCH
#CS SCK SDI LDAC
REFA DACA
VDD AVSS
C6 0.1uF
R8 100K
R7 1.5K
C8 0.01uF
11khz lpf
LPF
TM1POT THUMBPOT
VOLUME
AMP
TL072P
IC3A
IC3B
C9 100uF/6V
STEREOJACK
X1
JP11
SPK OUT
JP13
MMC_CS
POTSW
SW2
SW1
TM1SW THUMBPOT
LED2
R10 4.7K
WP2
LED1
R9 4.7K
WP1
LEDs
74125N-LARDO
IC4D
IC4P
Unused buffer
IC4C
IC4A
IC4B
R6 10K
WP
CD
CARD_DETECT
COMMON_SW
WRITE_PROTECT
DATA_OUT
DATA_IN
SCLK
CS
DAT1
DAT2
VDD
VSS1
VSS2
GND
SD & MMC
SD CARD
C5 0.1uF
C7 0.1uF
C2 0.1uF
C1 100uF/6V
IC1
IN OUT GND
C3 0.1uF
C4 100uF/6V
3.3V supply & bypass
Audio shield for Arduino
Released under CC 2.5, Attrib/Share-Alike
TITLE: waveshield11
Document Number:
http://www.ladyada.net/make/ashield
REV:
Date: not saved!
Sheet: 1/1