McGill | DP09

Rania Ouassif | Sansitha Panchadsaram | Anandamoyi Saha
Project Advisor Dr. Ioannis Psaromiligkos

Department of Electrical & Computer Engineering

Software-Controlled Analog Waveform Generator

Project Description

- Analog waveform generator with digitally controllable:
  - Amplitude
  - Frequency
  - Phase

Constraints

- Cost
  - Maximum budget of $100
- Scalability
  - Able to function well when adding new controlled parameters
- Ease of design
  - Can easily rebuild the circuit

Applications

- Educational institutions
- Response testing
- Signal generation
- Research and development
- Electrical repair

Market Research

- BitScope DWG100 & EVAL-AD9833SDZ
  - Software-controlled waveform generators

	Supply Voltage	Frequency	Amplitude	Phase	Cost
1. BitScope DWG 100	-12V to +12V	1Hz-1.5MHz	40mVpp-10Vpp	NA	$85
EVAL-AD9833 SDZ	2.3V to 5.5V	1Hz-12.5MHz	NA	0-180º	$111

Engineering Characteristics

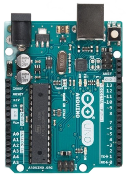
- Frequency : 100Hz-10kHz
  - Resolution: 0.1Hz
- Phase : 0° to 180°
  - Resolution: 2π/4096 ≈ 0.1°
- Amplitude : 2.5V-5V
  - Resolution: 0.16V
- Waveform types : Sine, Triangle, Square
- USB-Powered & two 9V batteries
- Input current: 10 μA
- Output current: 25 mA
- Digitally controlled through GUI capable of running on Linux, Windows, Mac

Challenges

- Components issues (e.g. overheated digital potentiometer and op amp)
- Adjusting voltage supply requirements for digital potentiometer and op amp

Design Process

ARDUINO UNO REV3



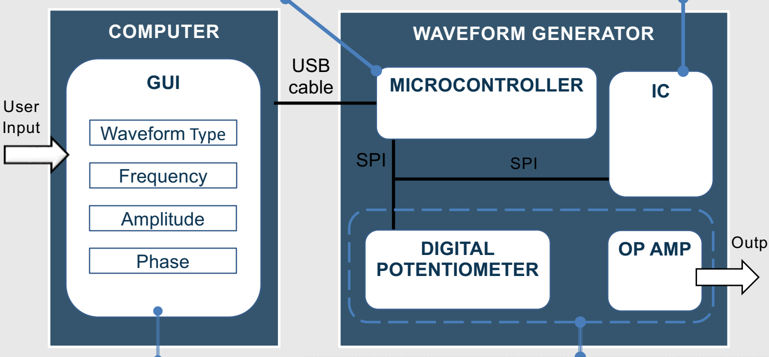
- Inexpensive
- Cross-platform
- Open source and extensible hardware and software

SPI Protocol

- Supports full-duplex communication
- Simple hardware connection
- No start and stop bits

AD9833

- DDS
- SPI protocol
- Software programmable frequency, phase and waveform type

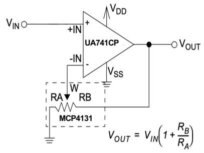


ARDUINO PROCESSING IDE

- Based on Java language
- Send/receive values directly from Arduino Board
- Can be extracted as an executable application

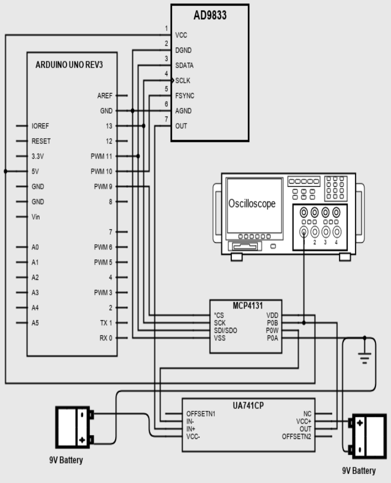
AMPLITUDE CONTROL

- Non-inverting programmable gain amplifier
- Built using MCP4131 digiPot and UA741CP op amp

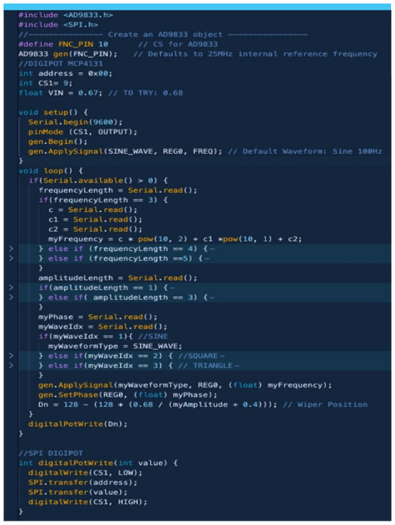


$$V_{OUT} = V_{IN}\left(1+\frac{R_B}{R_A}\right)$$

Schematic Diagram



Arduino Code

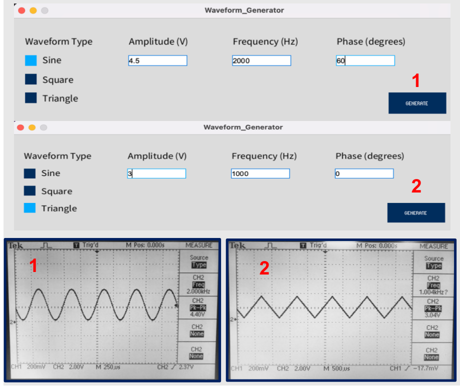


```
#include <AD9833.h>
#include <SPI.h>
//--------------- Create an AD9833 object ---------------
#define FNC_PIN 10       // CS for AD9833
AD9833 gen(FNC_PIN);   // Defaults to 25MHz internal reference frequency
//DIGIPOT MCP4131
int address = 0x00;
int CS1= 9;
float VIN = 0.67; // TO TRY: 0.68

void setup() {
  Serial.begin(9600);
  pinMode (CS1, OUTPUT);
  gen.Begin();
  gen.ApplySignal(SINE_WAVE, REG0, FREQ); // Default Waveform: Sine 100Hz
}
void loop() {
  if(Serial.available() > 0) {
    frequencyLength = Serial.read();
    if(frequencyLength == 3) {
      c = Serial.read();
      c1 = Serial.read();
      c2 = Serial.read();
      myFrequency = c * pow(10, 2) + c1 *pow(10, 1) + c2;
    } else if (frequencyLength == 4) {…
    } else if (frequencyLength ==5) {…
    }
    amplitudeLength = Serial.read();
    if(amplitudeLength == 1) {…
    } else if( amplitudeLength == 3) {…
    }
    myPhase = Serial.read();
    myWaveIdx = Serial.read();
    if(myWaveIdx == 1){ //SINE
      myWaveformType = SINE_WAVE;
    } else if(myWaveIdx == 2) { //SQUARE…
    } else if(myWaveIdx == 3) { // TRIANGLE…
    }
    gen.ApplySignal(myWaveformType, REG0, (float) myFrequency);
    gen.SetPhase(REG0, (float) myPhase);
    Dn = 128 - (128 * (0.68 / (myAmplitude + 0.4))); // Wiper Position
  }
  digitalPotWrite(Dn);
}

//SPI DIGIPOT
int digitalPotWrite(int value) {
  digitalWrite(CS1, LOW);
  SPI.transfer(address);
  SPI.transfer(value);
  digitalWrite(CS1, HIGH);
}
```

Testing

- **Subsystem Testing**
  1. AD9833 breakout board
  2. MCP4131 digiPot
  3. UA741CP op amp
  4. GUI
- **Integration Testing**
  1. UA741CP op amp & MCP4131 digiPot
  2. AD9833 breakout board & GUI
  3. Entire system

Results



Waveform_Generator

Waveform Type	Amplitude (V)	Frequency (Hz)	Phase (degrees)
■ Sine	4.5	2000	60
■ Square			
■ Triangle			

1 GENERATE

Waveform_Generator

Waveform Type	Amplitude (V)	Frequency (Hz)	Phase (degrees)
■ Sine	3	1000	0
■ Square			
■ Triangle			

2 GENERATE

1
Tek Trig'd M Pos: 0.000s MEASURE
Source Type
CH2 Freq 2.000kHz
CH2 Pk-Pk 4.40V
CH2 None
CH2 None
CH1 200mV CH2 2.00V M 250μs CH2 2.37V
2
Tek Trig'd M Pos: 0.000s MEASURE
Source Type
CH2 Freq 1.004kHz?
CH2 Pk-Pk 3.04V
CH2 None
CH2 None
CH1 200mV CH2 2.00V M 500μs CH1 -17.7mV

Conclusion

- Built a cost-effective software-controlled waveform generator
- Total cost: $50

Lessons learnt:

- How to understand trade-offs
- Time management

Retrospect:

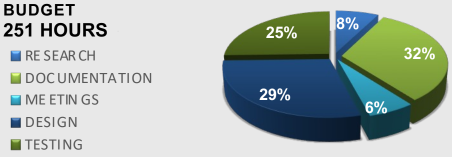
- Achieved more tasks in parallel individually
- Thoroughly read specifications in the datasheets
- Account for any potential hardware/software failures

Future Plans

- Implement additional parameters
- Implement additional waveform types
- Add a second waveform channel
- Find alternative to 9V batteries
- Improve accuracy of amplitude
- Reduce noise
- Increase range of frequency and amplitude

Project Management

BUDGET
251 HOURS



RESEARCH
DOCUMENTATION
MEETINGS
DESIGN
TESTING