The trials and tribulations of building a phase-sensitive detector with an Arduino microcontroller

K. D. Schultz

Department of Physics
Hartwick College

MAS-APS, 2014

Outline

Original Goals

- Use Arduino as a tool for teaching about phase-sensitive detection, like.....

Original Goals

- Use Arduino as a tool for teaching about phase-sensitive detection, like.....
- To do so with only the Arduino, a computer for display purposes, and passive external components (resistors and capacitors)

Original Goals

- Use Arduino as a tool for teaching about phase-sensitive detection, like.....
- To do so with only the Arduino, a computer for display purposes, and passive external components (resistors and capacitors)

Original Goals

- Use Arduino as a tool for teaching about phase-sensitive detection, like.....
- To do so with only the Arduino, a computer for display purposes, and passive external components (resistors and capacitors)
- Can be done, but different lessons are learned

Why PSD?

- Phase Sensitive Detection (PSD) is the basis of many techniques in physics and engineering

Why PSD?

- Phase Sensitive Detection (PSD) is the basis of many techniques in physics and engineering
  - Homodyne detection

Why PSD?

- Phase Sensitive Detection (PSD) is the basis of many techniques in physics and engineering
  - Homodyne detection
  - Interferometry

Why PSD?

- Phase Sensitive Detection (PSD) is the basis of many techniques in physics and engineering
  - Homodyne detection
  - Interferometry
  - Lock-in amplifiers

Why PSD?

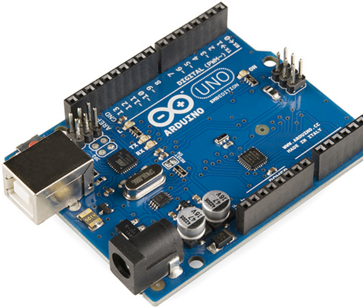
- Phase Sensitive Detection (PSD) is the basis of many techniques in physics and engineering
  - Homodyne detection
  - Interferometry
  - Lock-in amplifiers
- Black boxes are useful for application work, but not so much for pedagogical purposes

Why PSD?

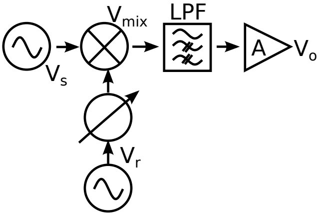
- Phase Sensitive Detection (PSD) is the basis of many techniques in physics and engineering
  - Homodyne detection
  - Interferometry
  - Lock-in amplifiers
- Black boxes are useful for application work, but not so much for pedagogical purposes
- Software PSD allows students to peek into the black box.

Why Arduino?

- Cheap
- Popular
  - Lots of support
- Simple programming environment
  - Perhaps too simple, IDE has very poor debugging tools.
  - Really no debugging tools, except for serial out
- Works well with Processing, which is a free and powerful language for visualization



PSD Basics



Mathematics of PSD

$$V_{mix} = V_s V_r \left[\left(\cos\left(\omega_s - \omega_r\right) t - \left(\phi_s - \phi_r\right) \right] \right.$$

$$V_o = A \frac{V_s V_r}{2} \left[\cos(\phi_s - \phi_r) \right]$$

Restrictions

- $\omega_r = \omega_s$
- V_s and V_r have no DC offset

What do we need? Can we do it?

Needed features

1. Generate reference signal.
2. Adjustable phase to maximize signal.
3. Get signal into Arduino
4. Mix signal and reference and filter the results
5. Display results

What do we need? Can we do it?

Needed features

1. Generate reference signal.
2. Adjustable phase to maximize signal.
3. Get signal into Arduino
4. Mix signal and reference and filter the results
5. Display results

Issues

1. Arduino has no analog out

What do we need? Can we do it?

Needed features

1. Generate reference signal.
2. Adjustable phase to maximize signal.
3. Get signal into Arduino
4. Mix signal and reference and filter the results
5. Display results

Issues

1. Arduino has no analog out
2. Shifting phase is difficult to do internally

What do we need? Can we do it?

Needed features

1. Generate reference signal.
2. Adjustable phase to maximize signal.
3. Get signal into Arduino
4. Mix signal and reference and filter the results
5. Display results

Issues

1. Arduino has no analog out
2. Shifting phase is difficult to do internally
3. Positive voltages only

What do we need? Can we do it?

Needed features

1. Generate reference signal.
2. Adjustable phase to maximize signal.
3. Get signal into Arduino
4. Mix signal and reference and filter the results
5. Display results

Issues

1. Arduino has no analog out
2. Shifting phase is difficult to do internally
3. Positive voltages only
4. Limited variable memory on Arduino

What do we need? Can we do it?

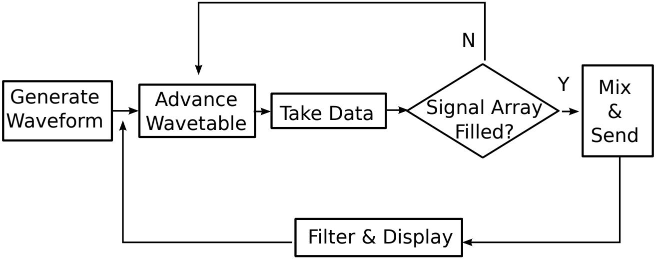
Needed features

1. Generate reference signal.
2. Adjustable phase to maximize signal.
3. Get signal into Arduino
4. Mix signal and reference and filter the results
5. Display results

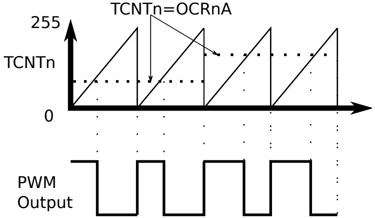
Issues

1. Arduino has no analog out
2. Shifting phase is difficult to do internally
3. Positive voltages only
4. Limited variable memory on Arduino
5. Need to use serial over USB to Processing

Flowchart


Generate
Waveform
Advance
Wavetable
Take Data
Signal Array
Filled?
N
Y
Mix
&
Send
Filter & Display

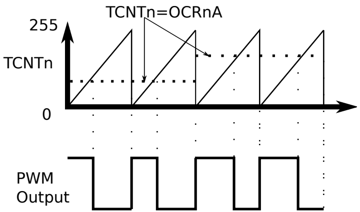
Creating a Reference Signal



Timers and Interrupts Part I

- The ratio of PWM on to off determines an average “DC” signal
- When register TCNT1 reaches OCR1A PWM goes low
- When TCNTn overflows PWM goes high again

Creating a Reference Signal



$$f_{ref} = \frac{\text{TCNT2 rate}}{\text{OCR2A value} \times \text{wavetable length}}$$

Timers and Interrupts Part II

- Need fast timer2 and regular timer1, which outputs PWM
- When timer2 reaches OCR2A:
  - Update OCR1A from wavetable
  - read signal at AnalogIn
- When timer1 counts up to OCR1A, PWM goes low
- When timer1 overflows PWM goes high again

Signal Input

Phase Manipulation

Display

For Further Reading I

A. Author.
Handbook of Everything.
Some Press, 1990.

S. Someone.
On this and that.
Journal of This and That, 2(1):50–100, 2000.