Application Report
SLVA416–September 2010

High Resolution Microstepping Driver With the DRV88xx Series

ABSTRACT

This document is provided as a supplement to the DRV8812/13 and DRV8828/29 datasheets. It details the hardware implementation of the DRV8812/13/28/29 and an MSP430F1612 microcontroller in a combination that allows the dual DC motor driver to drive a bipolar stepper motor in microstepping configuration with increased degrees of microstepping.

Contents

List of Figures

List of Tables

Introduction

Bipolar stepper motors are often designed to be driven at full steps. However, it is a known fact that when each step is divided into multiple and smaller steps (referred to as micro steps), better motion quality can be observed. The application of microstepping to practically any stepper application allows for other improvements such as:

1. Better torque response
2. Less vibration
3. Less resonance incidence

Any stepper can be microstepped. TI's DRV8811, DRV8821 will allow users to apply microstepping to bipolar stepper motors with up to 8 degrees of microstepping. The DRV8824 and DRV8825 will go 2 steps further, offering both 16 and 32 degrees of microstepping, as well as 8th, quad and half. But what if the application requires an even finer resolution? What if the application requires resolutions in the order of 64, 128 or even 256 degrees of microstepping?

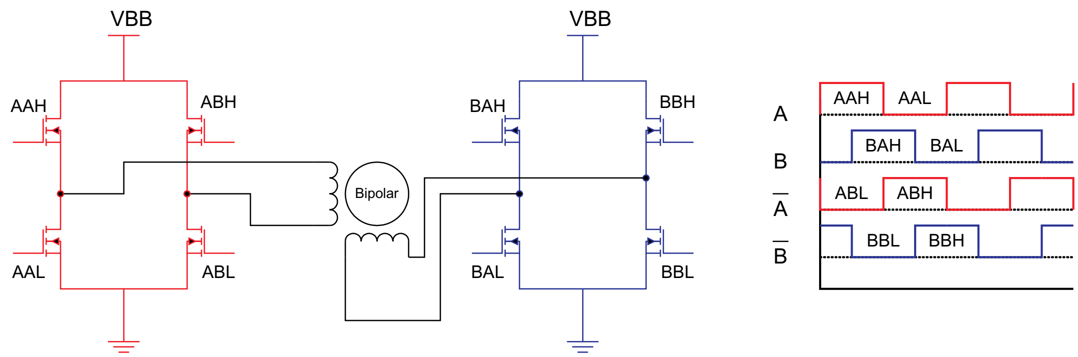
Thanks to the versatility included within the DRV8812/13 devices, this can be easily achieved by modulating the independent VREF analog inputs. All which needs to be done is to apply an analog waveform to said reference voltage inputs, and the winding current magnitude will be regulated to follow the respective input. Generating this waveform can be accomplished by the utilization of a DAC output such as the ones found on an MSP430 microcontroller. Putting all of this together gives us a series of advantages such as:

1. 1. Degrees of microstepping can be made as large as there is resolution on the DAC block. With a 12 bit DAC, you could have up to 4096 levels of microstepping. Far much more than most applications should require.
2. 2. Any waveform, however intricate, can be encoded into the VREF analog inputs. This could be used to resolve any magnetic non linearity within the stepper motor itself.

In this application note we will detail the mechanisms put in place to achieve high resolution microstepping by utilizing a microcontroller, such as the MSP430F1612, in combination with the DRV8812/13 device. The same concepts would apply if different drivers are utilized. Per example, for larger current drive, two DRV8828/29 devices could be used.

Full Step vs Microstep

1 The Full Step



Bipolar Motor and Drive

Polarization Sequencing

Figure 1. Full Step

Stepper motors are brushless motors. In other words, they lack a brush making mechanical contact with a commutator, as housed on the typical brushed DC motor. Because of this, stepper motors must be electronically commutated. The most common way of commutating the rotor into motion is by modulating the magnetic field of each stator electro magnet.

There are two stator electromagnets on any conventional stepper motor, which is why two H Bridges are needed. The DRV8812 offers the capability to drive both of these motor windings. By modulating both PHASE A and PHASE B pins, to obtain a polarization waveform such as the one depicted in Figure 1, the motor will move in full steps.

2 The Micro Step

If full steps are all that the application requires, then Figure 1 is the optimal solution. However, the great majority of applications can certainly benefit from dividing full steps into smaller steps.

Figure 2 shows two full steps on the top and a way to divide each full step into 8 smaller steps. We have chosen a sine wave shape, although in reality this waveform can have any shape the users chooses. The idea behind microsteps is to make the current flowing through the winding different than FULL ON on one direction and FULL ON on the opposite direction. This allows for the rotor to be pushed/pulled in increments rather than full force.

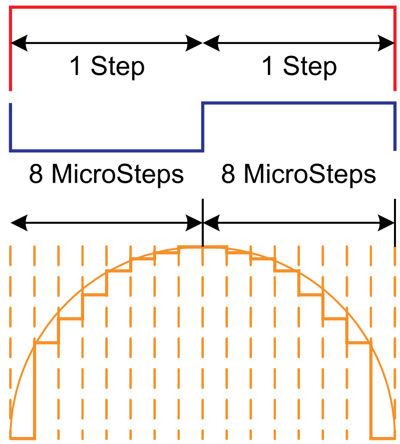


Figure 2. Micro Step

Because the DRV8812 has the capability to regulate winding current, and said current is directly proportional to the VREF input, it is possible to embed the prospective waveform at the current side. Figure 3 shows the result of embedding said waveform into the winding current, which is the concept behind this application note.

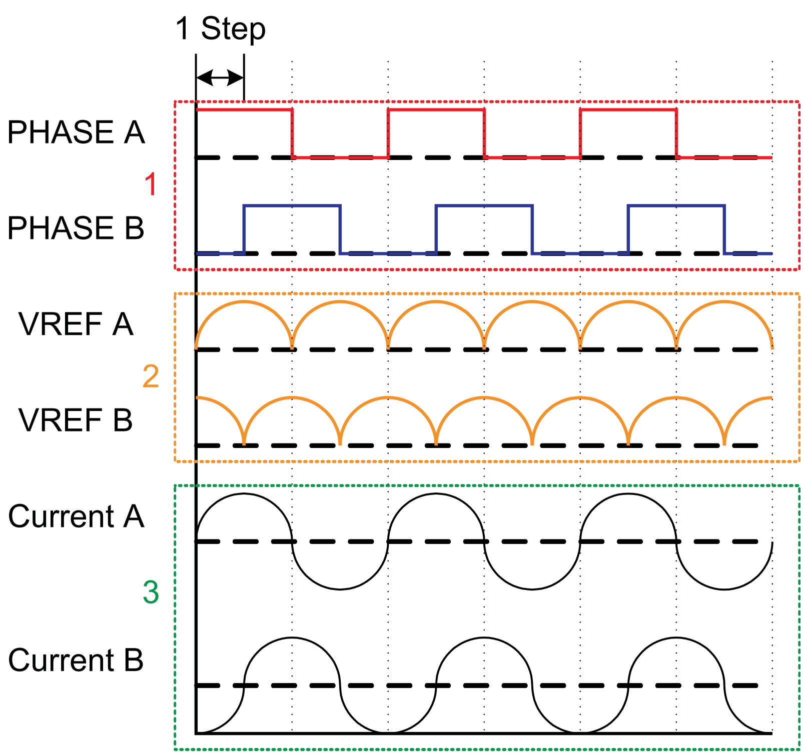


Figure 3. Embedded Waveform Into Winding Current

A microcontroller, such as the MSP430F1612 and utilizing GPIO outputs, will drive the DRV8812's PHASE inputs in the alternating fashion we saw on previous figures. However, and at the same time, the VREF pins will be modulated by using the two DAC outputs to superimpose a sine wave crest.

As a result, the motor windings see a full sine wave of current being applied to them. Because the current is segmented in smaller steps, the rotor also moves in smaller steps.

Application Block Diagram

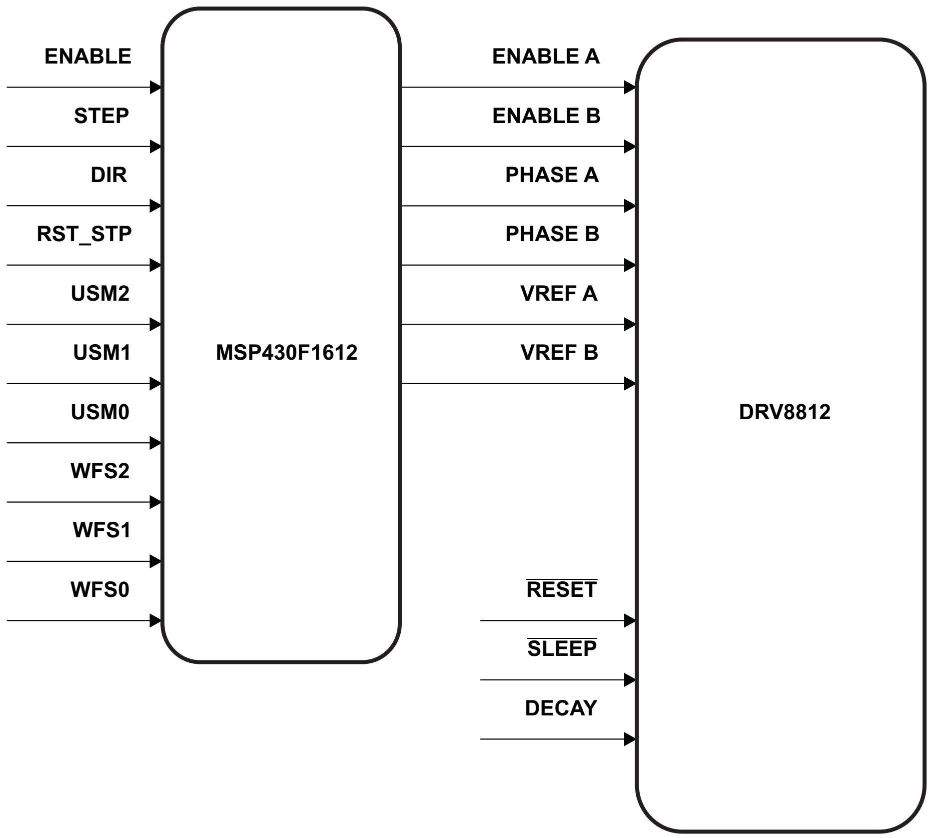


Figure 4. Application Block Diagram

In essence, the combination portrayed in Figure 4 is similar to how a stepper with internal indexer (such as the DRV8811 or DRV8821) operates. A logic block samples a series of inputs and generates control signals to the dual H Bridge device. Typical internal indexer driver devices represent an encased application and can not be modified. However, by splitting the application into a microcontroller and a driver, more flexibility is obtained.

Figure 4 shows how the MSP430F1612 drives the control signals such as PHASE, ENABLE and VREF. ENABLE and PHASE signals are derived from conventional GPIO configured as outputs, whereas the VREF analog signals are derived from both of MSP430F1612's DAC outputs. An interrupt subroutine computes the next step and generates the next microstep.

1 Microstepping Indexer

At the heart of our microstep generation engine is a lookup table that houses up to eight waveforms, In reality, the MSP430F1612 has enough space to hold much more than eight tables, but for this application note and demonstration purposes, eight tables seemed enough.

Each one of these lookup tables is 512 steps deep. Which waveform table is selected is a factor of the three WFS bits.

To generate a step, two values are read from the table and loaded into the DAC registers. In order for PHASE A to be 90 degrees out of phase with regards to PHASE B, each table item read is located on addresses 256 steps apart from each other (when on 256 microsteps mode). Whether PHASE A lags or leads B depends on the DIR bit. For Clockwise, we chose PHASE A leads B, while for counter clockwise, PHASE A lags PHASE B was chosen. This is totally arbitrary and either way is perfectly acceptable as long as the motor is wired accordingly.

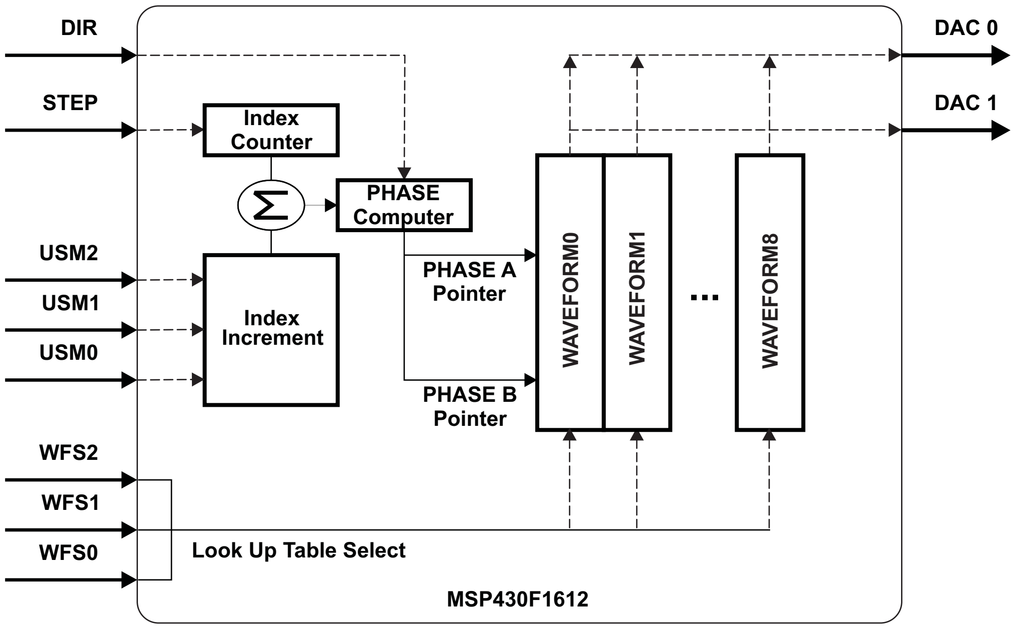


Figure 5. Microstepping Indexer

Every time a STEP is issued, a counter is incremented. This counter becomes a pointer to the lookup table. The counter is incremented by 1 if on the highest resolution setting. In this case, maximum degrees of microstepping is obtained. However, in order to obtain less degrees of microstepping, there is no need to have a smaller look up table as the same large look up table suffices.

In order to obtain lesser degrees of microstepping, the counter is incremented by a larger factor, that must be base 2. Per example, to obtain 256 degrees of microstepping, the incrementing value is 1 and to obtain 128 degrees of microstepping, the incrementing factor is 2. Subsequent multiplications of 2 for the incrementing factor, take the microstepping degrees down to half of what they were before.

The Index Increment is a factor of the USMx bits. With three bits we have up to 8 different combinations. In this case, the indexer can operate with half step, quad step as well as 8, 16, 32, 64, 128, 256 degrees of microstepping, as depicted in Table 1.

Table 1. Microstepping Operation

Degrees of Microstepping	Divider Factor (Lookup Table Index Increment)
256	1
128	2
64	4
32	8
16	16
8	32
4	64
2	128

2 Microcontroller Side Control Signals

ENABLE: Microcontroller timer input configured to capture and interrupt on both edges. Signals the microcontroller whether the DRV8812 H Bridges should be powered up or not. Microcontroller remains powered on low power mode when ENABLE is LO and will operate accordingly when ENABLE is HI.

STEP: Microcontroller timer input configured to capture and interrupt on rising edges. A transition from LO to HI tells the microcontroller to issue a new step. Transitions from HI to LO are ignored, although it is the user's prerogative to code this so that both transitions generate a step.

DIR: Microcontroller timer input configured to capture and interrupt on both edges. Selects the bipolar motor direction of rotation. LO implies counter clockwise rotation, while HI selects clockwise rotation. This was arbitrarily chosen and can be changed according to preference.

STP_RST: Microcontroller timer input configured to capture and interrupt on both edges. Clears the internal indexer and disables the H Bridge. When STP_RST is made LO, the indexer look up table is returned to pointer 0 and the H Bridge is disabled. When STP_RST returns to HI, the H Bridges are enabled and motion starts from lookup table position 0.

USMx: Microcontroller timer input configured to capture and interrupt on both edges. Selects the 8 possible degrees of microstepping resolution. In this application note, half step, quad step, as well as 8, 16, 32, 64, 128 and 256 degrees of microstepping were coded.

WFSx: Microcontroller interrupt input (GPIO). Waveform Select bits. Select from the eight internal look up tables.

3 Microstepping Engine Algorithm

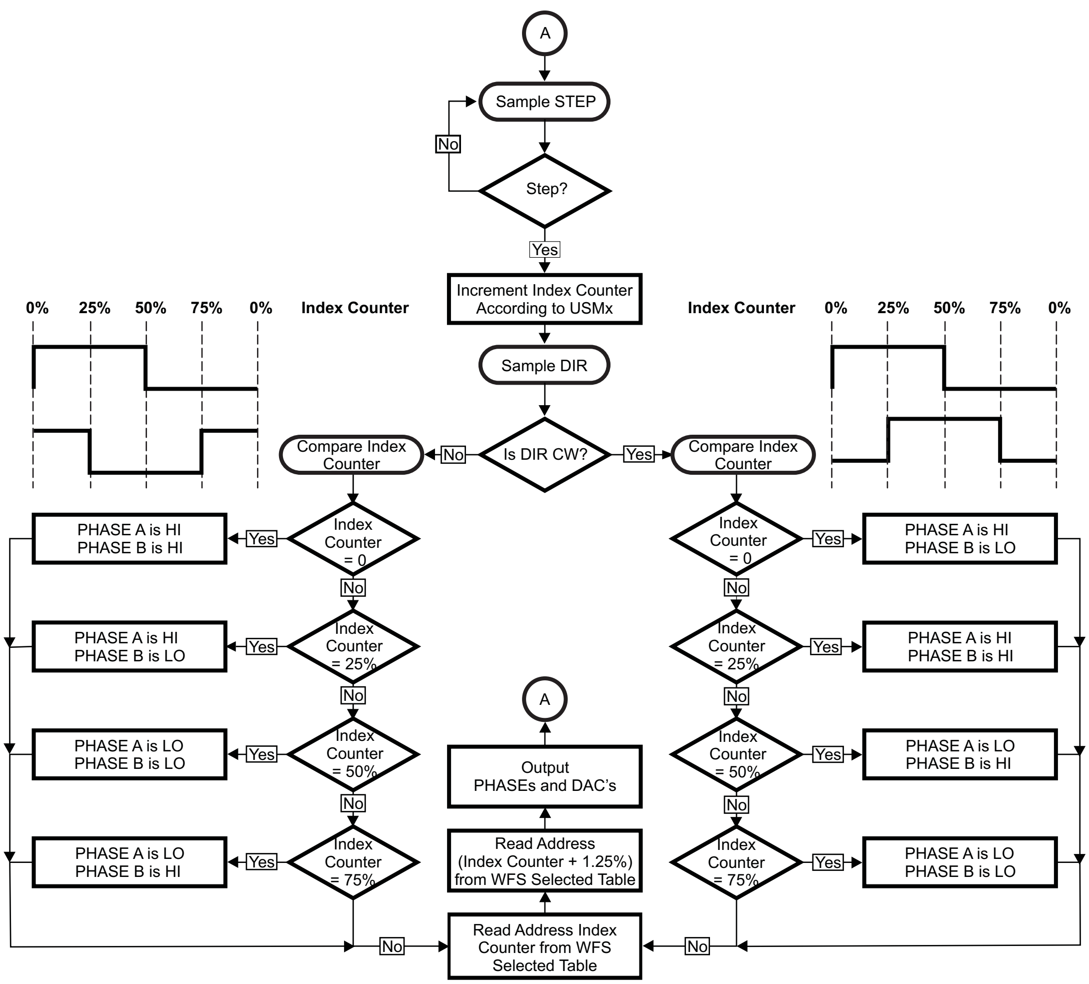


Figure 6. Microstepping Engine Algorithm

Example Plots

On the following pictures, the Channel Legend applies:

- CH1: VREF A
- CH2: PHASE A Pin
- CH3: Winding A Current

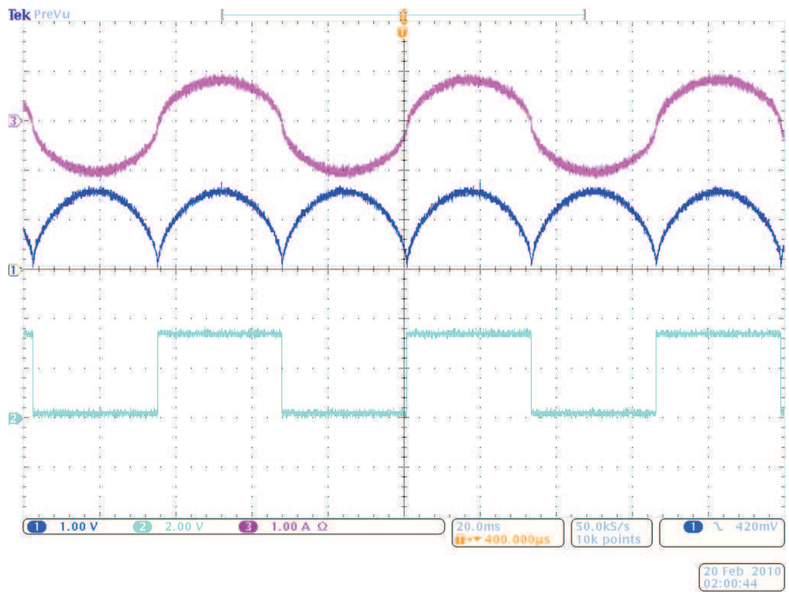


Figure 7. Squared Sine Wave With 256 Degrees of Microstepping

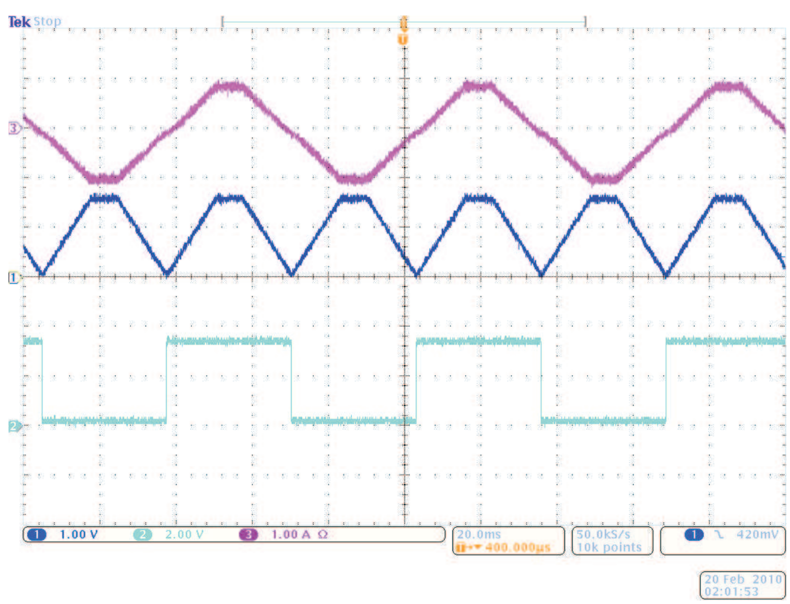


Figure 8. Trapezoid Waveshape With 256 Degrees of Microstepping

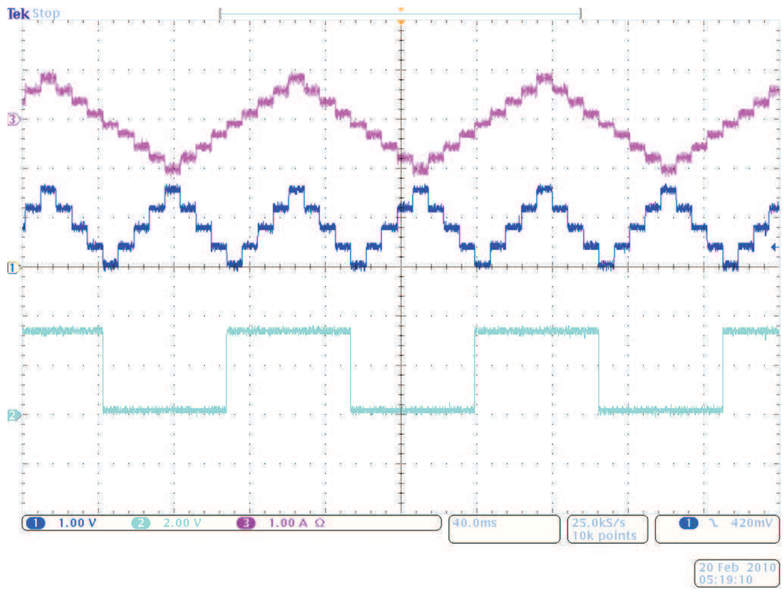
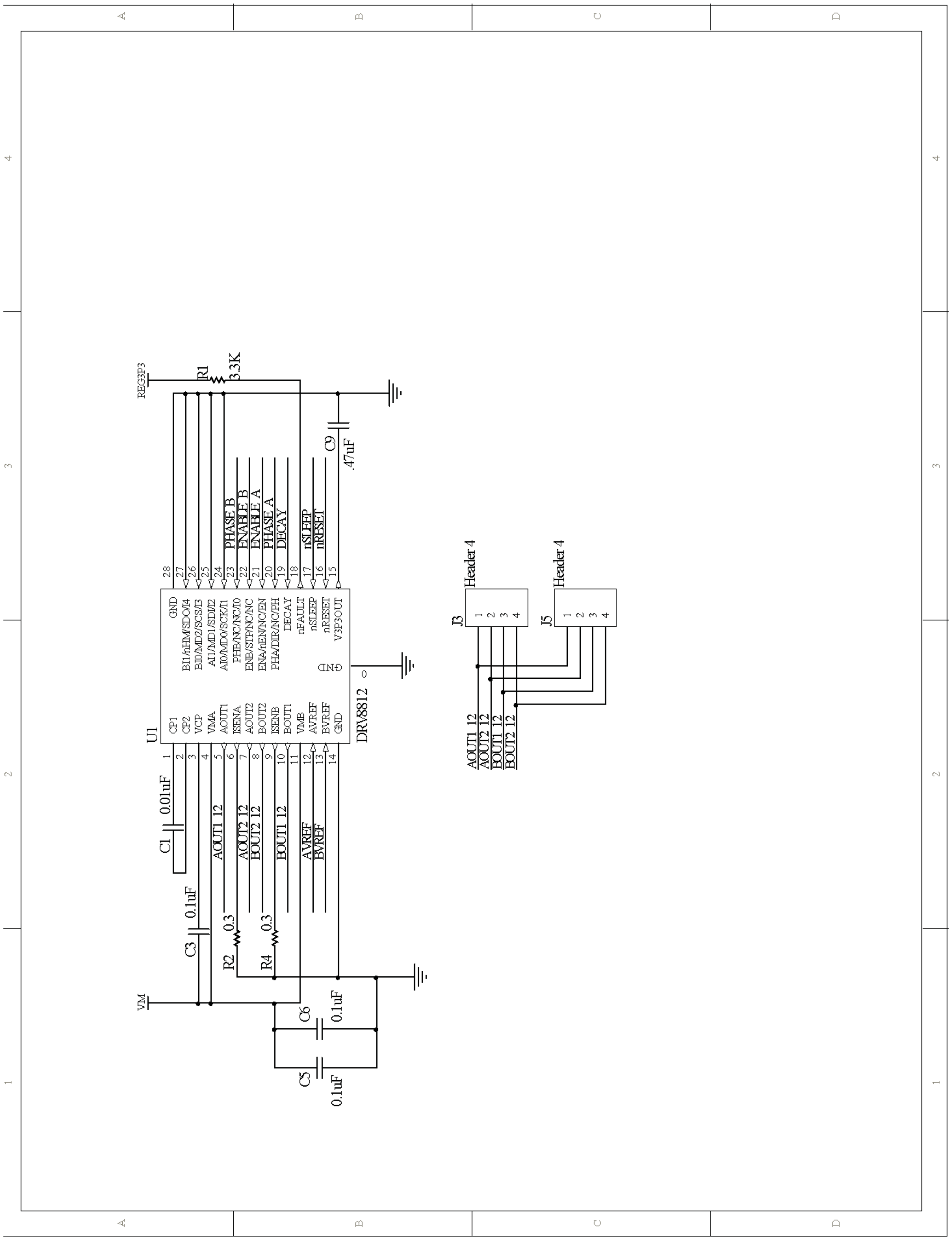
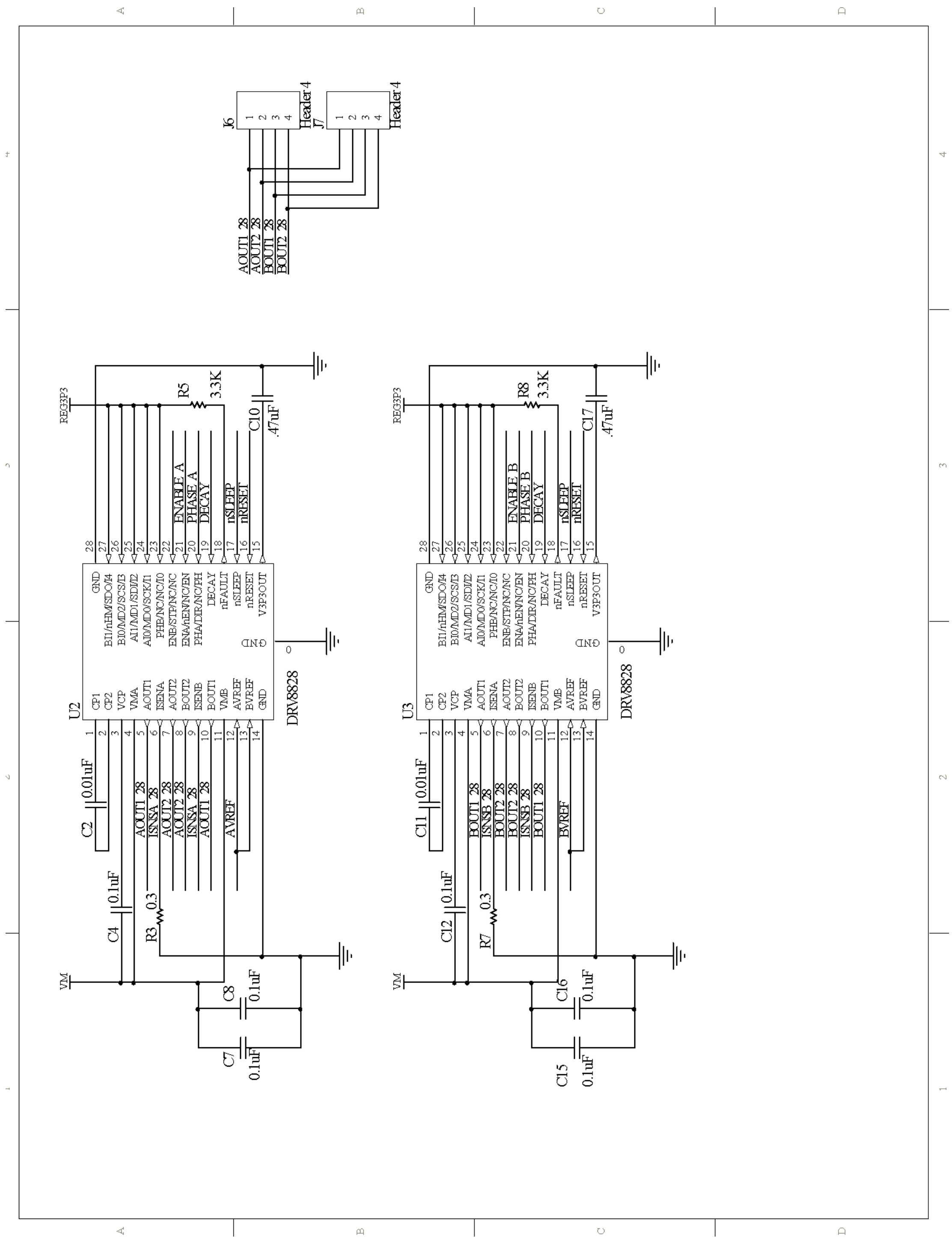
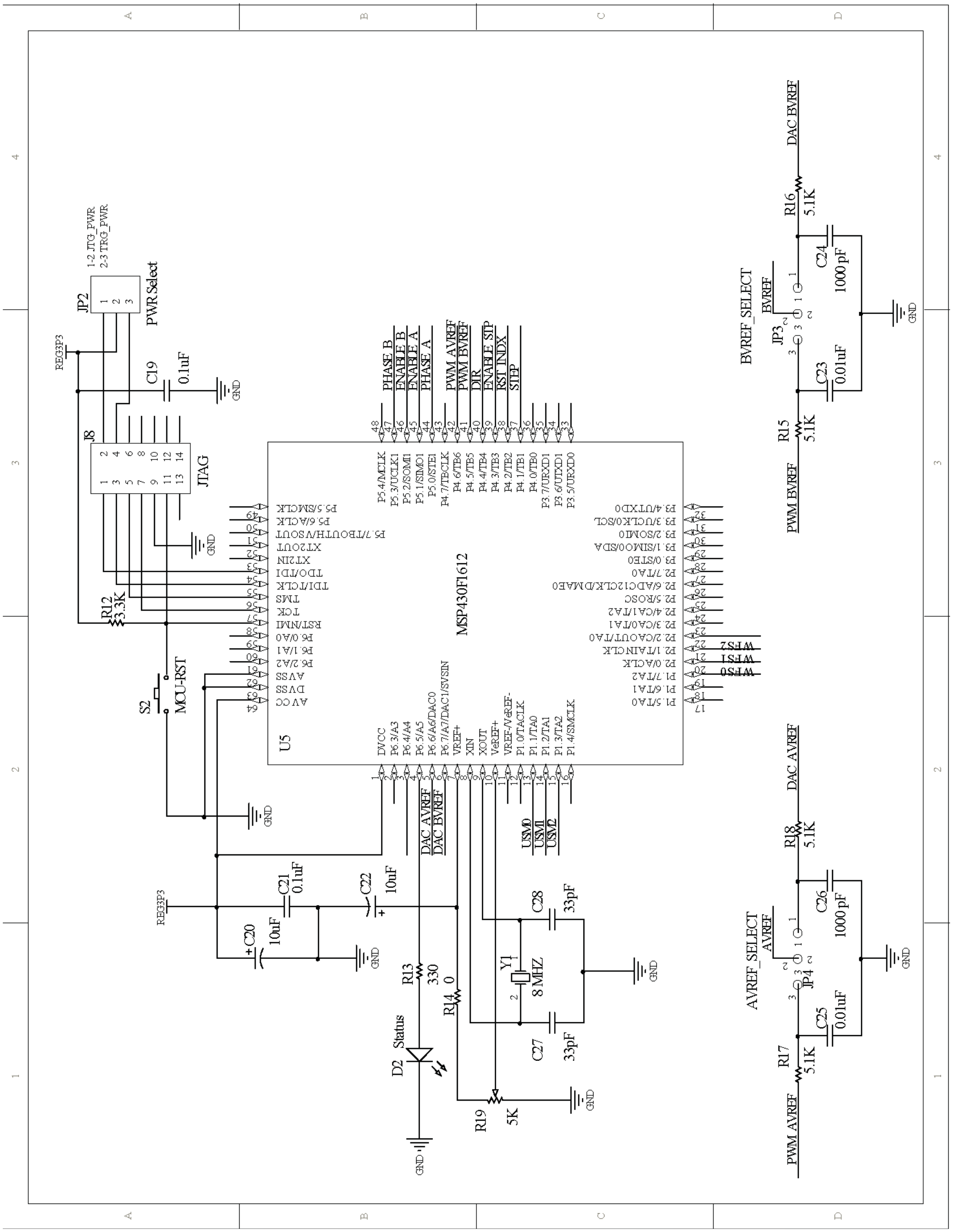


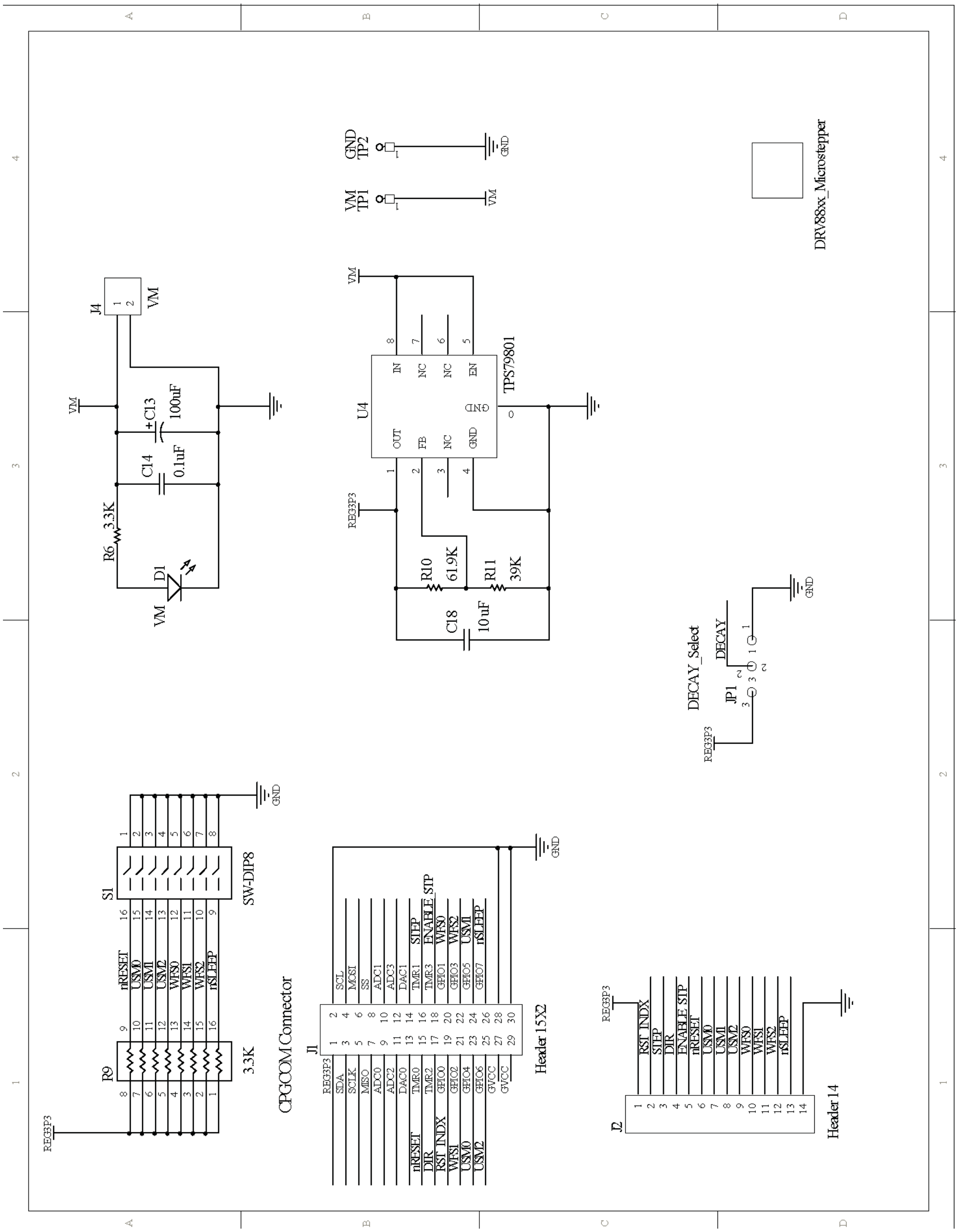
Figure 9. Triangular Waveshape With 4 Degrees of Microstepping

Schematic


REG3P3
R1
3.3K
C9
.47uF
PHASE B
ENABLE B
ENABLE A
PHASE A
DECAY
nSLEEP
nRESET
U1
DRV8812
GND
BI1/nHM/SDO/I4
BI0/MD2/SCS/I3
AI1/MD1/SDI/I2
AI0/MD0/SCK/I1
PHB/NC/NC/I0
ENB/STP/NC/NC
ENA/nEN/NC/EN
PHA/DIR/NC/PH
DECAY
nFAULT
nSLEEP
nRESET
V3P3OUT
CP1
CP2
VCP
VMA
AOUT1
ISENA
AOUT2
BOUT2
ISENB
BOUT1
VMB
AVREF
BVREF
GND
C1
0.01uF
C3
0.1uF
AOUT1 12
AOUT2 12
BOUT2 12
BOUT1 12
AVREF
BVREF
R2
0.3
R4
0.3
VM
C6
0.1uF
C5
0.1uF
J3
Header 4
J5
Header 4


J6
Header 4
J7
Header 4
AOUT1 28
AOUT2 28
BOUT1 28
BOUT2 28
REG3P3
R5
3.3K
C10
.47uF
ENABLE A
PHASE A
DECAY
nSLEEP
nRESET
U2
DRV8828
C2
0.01uF
C4
0.1uF
R3
0.3
VM
C8
0.1uF
C7
0.1uF
AVREF
R8
3.3K
C17
.47uF
ENABLE B
PHASE B
U3
C11
C12
R7
C16
C15
BVREF


MSP430F1612
U5
J8
JTAG
JP2
PWR Select
1-2 JTG_PWR
2-3 TRG_PWR
REG3P3
C19
0.1uF
GND
R12
3.3K
S2
MCU-RST
C21
0.1uF
C22
10uF
C20
10uF
R13
330
R14
0
D2
Status
Y1
8 MHZ
C27
33pF
C28
33pF
R19
5K
DAC AVREF
DAC BVREF
USM0
USM1
USM2
PHASE B
ENABLE B
ENABLE A
PHASE A
PWM AVREF
PWM BVREF
DIR
ENABLE STP
RST INDX
STEP
WFS2
WFS1
WFS0
BVREF_SELECT
BVREF
JP3
C24
1000pF
C23
0.01uF
R16
5.1K
R15
5.1K
PWM BVREF
DAC BVREF
AVREF_SELECT
AVREF
JP4
C26
1000pF
C25
0.01uF
R18
5.1K
R17
5.1K
PWM AVREF
DAC AVREF


DRV88xx_Microstepper
TPS79801
U4
TP1
TP2
J4
R6 3.3K
D1
C13 100uF
C14 0.1uF
R10 61.9K
R11 39K
C18 10 uF
DECAY_Select
JP1
S1
SW-DIP8
R9
3.3K
J1
CPGCOM Connector
Header 15X2
J2
Header 14

PCB Layout

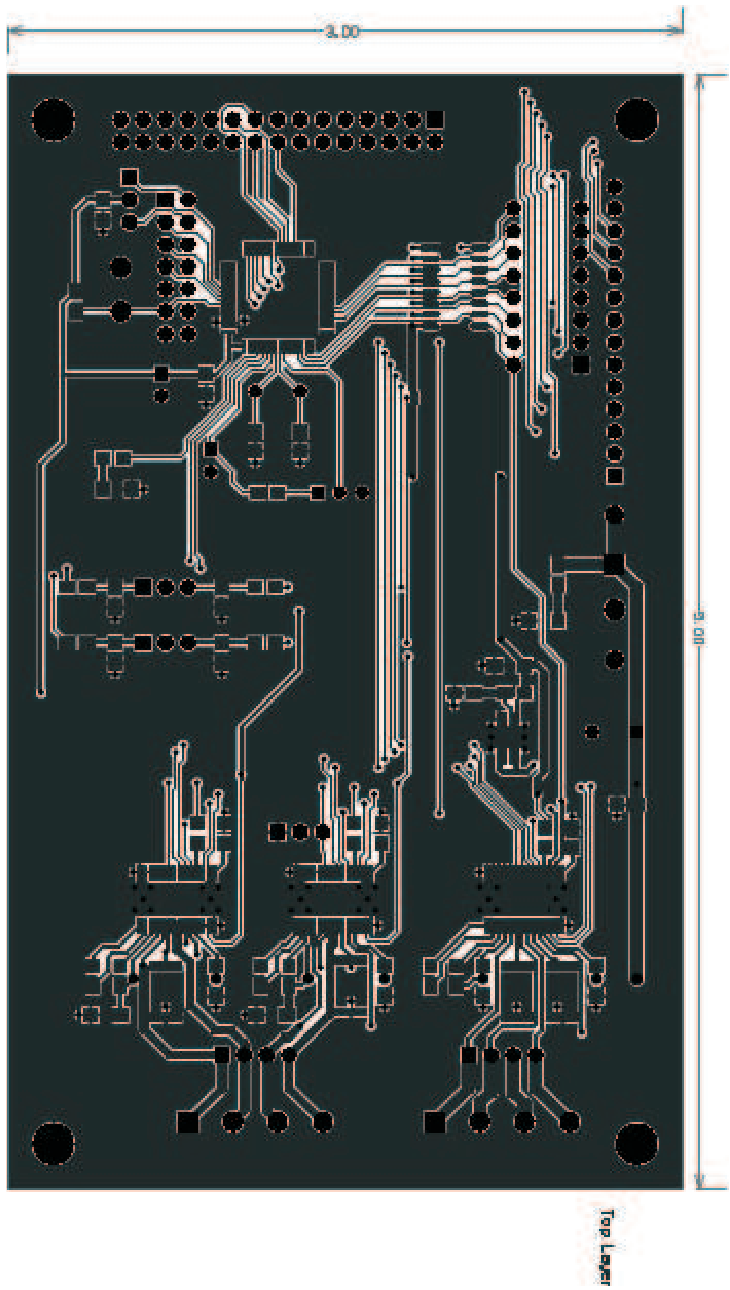


Figure 10. Top Layer

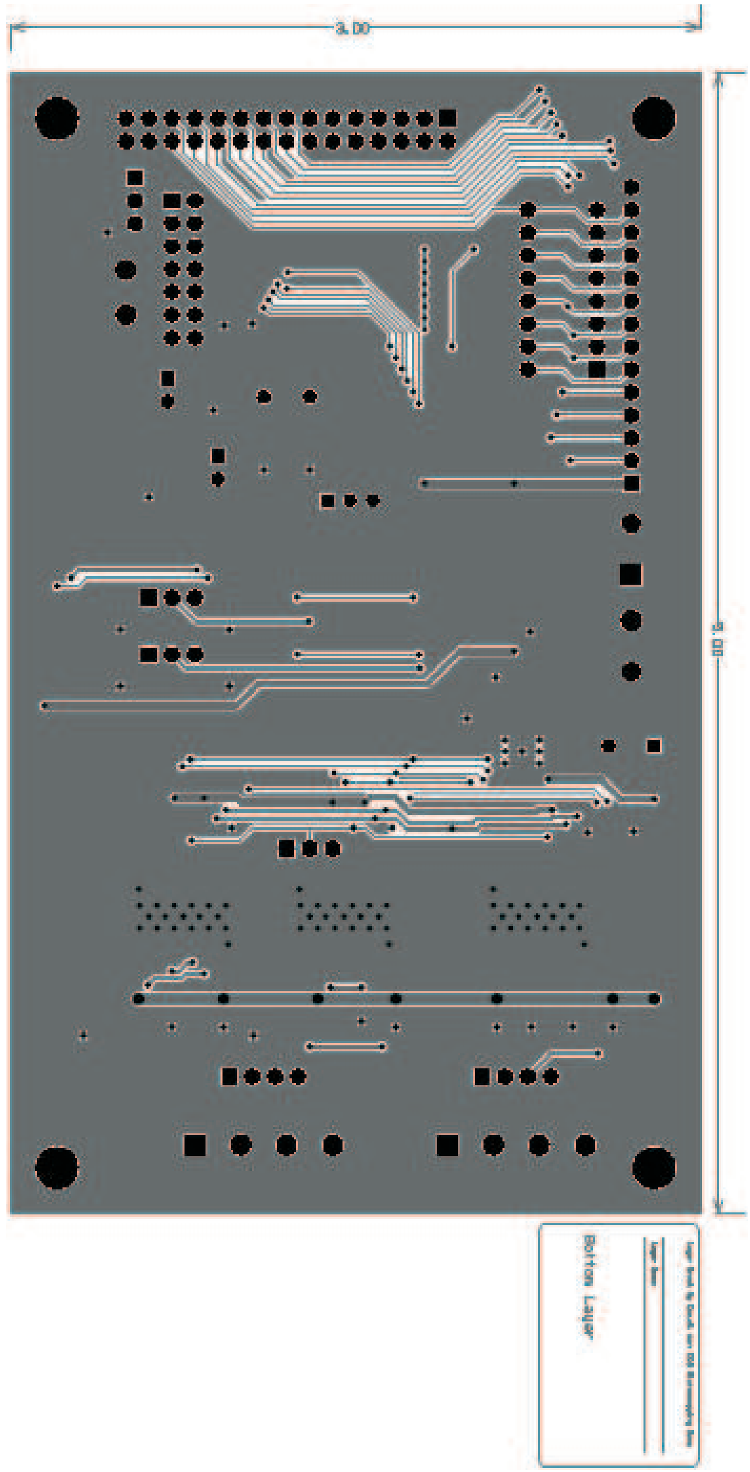
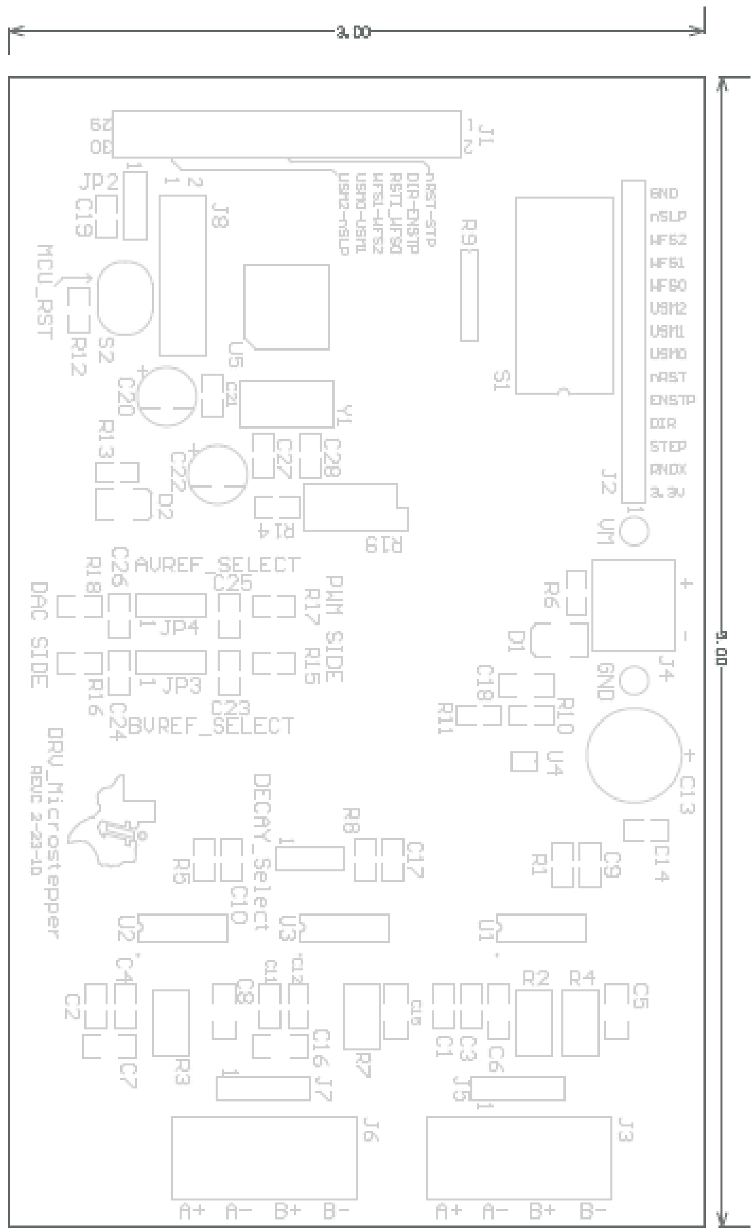


Figure 11. Bottom Layer


3.00