SEC-EPS-12V-APM19-GEVB: Electronic Power Steering (EPS) with APM19 Reference Design

TND6376/D

Device	Application	Input Voltage	Max Current	Topology
NXV04V120DB1 FDBL9406−F085T6 LV8968BB NCV7340 NCV890200 NCV20072 NCV20074 NCV7805 …	Electronic Power Steering	9−20 Vdc	80 A	FOC Algorithm for Motor Driver

PHOTOGRAPH OF THE EVALUATION BOARD

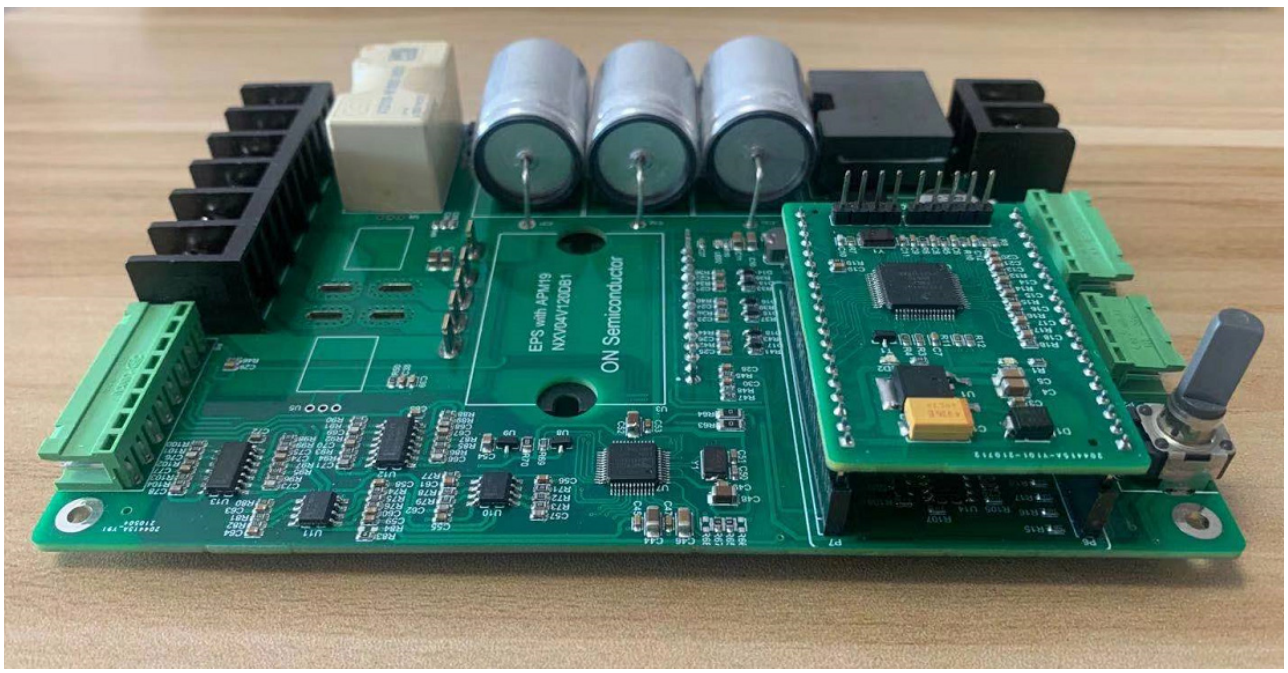

Figure 1. Photograph of the Evaluation Board

KEY FEATURES

Whole Solution:

- 3−phase inverter Automotive Power Module (APM) for PMSM.
- 3−phase pre−driver.
- CAN communication.
- DC−DC/LDO and amplifier.
- Integrated software algorithm.

Solution Features:

- Uses the APM19 (NXV04V120DB1) for PMSM. Automotive Power Modules (APMs) enable the design of small, efficient and reliable systems.
- Simplified vehicle assembly.
- Low thermal resistance.
- Uses 3 half−bridge gate−drivers (LV8968BB).
  - 8 V−28 V supply voltage capable.
  - Extensive system protection.
  - 3.3 V and 5 V input logic compatible.
- FOC & current−loop to drive motor, providing fast response and high efficiency.

Component Features:

NXV04V120DB1:

- Three−phase inverter bridge for variable speed motor drive.
- RC snubber for low EMI.
- Current sensing and temperature sensing.
- Electrically isolated DBC substrate for low thermal resistance.
- AEC qualified − AQG324.

LV8968BB:

- Full drive power from 8V to 28V supply voltage, with transient tolerance from 4.5 V to 40 V.
- Up to 25 kHz motor PWM with individual six gate control or Drive−3 mode with integrated programmable dead−time.
- Extensive system protection features including:
  - Drain−source short detection for external MOSFET.
  - Overcurrent shutoff.
  - Low gate voltage warning.
  - Over−temperature warning and shutoff.
  - Over / undervoltage protection.
- Supports functional safety design level ASIL−B.
- AEC−Q100 qualified and PPAP capable.

NCV20072 & NCV20074:

- Wide supply range: 2.7 V to 36 V.
- Rail−to−Rail Output.
- Wide bandwidth: 3 MHz typical at VS = 2.7 V.
- High slew rate: 2.8 V/μs typical at VS = 2.7 V.
- Low supply current: 405 μA per channel at VS = 2.7 V.
- Wide temperature range: −40°C to 125°C.
- NCV Prefix for automotive and other applications requiring unique site and control change requirements; AEC−Q100 qualified and PPAP capable.

NCV890200:

- 2 MHz free−running switching frequency.
- Typical 4.5 V to 18 V automotive input voltage range.
- Short−circuit protection enhanced by frequency fold−back.
- ±1.75% output voltage tolerance.
- 2.2 A (min) cycle−by−cycle peak current limit.
- NCV prefix for automotive and other applications requiring unique site and control change requirements; AEC−Q100 qualified and PPAP capable.

SCHEMATICS AND CIRCUIT DESCRIPTION

The system diagram is Figure 2. The key **onsemi** elements of the EPS reference design are marked in the orange blocks.

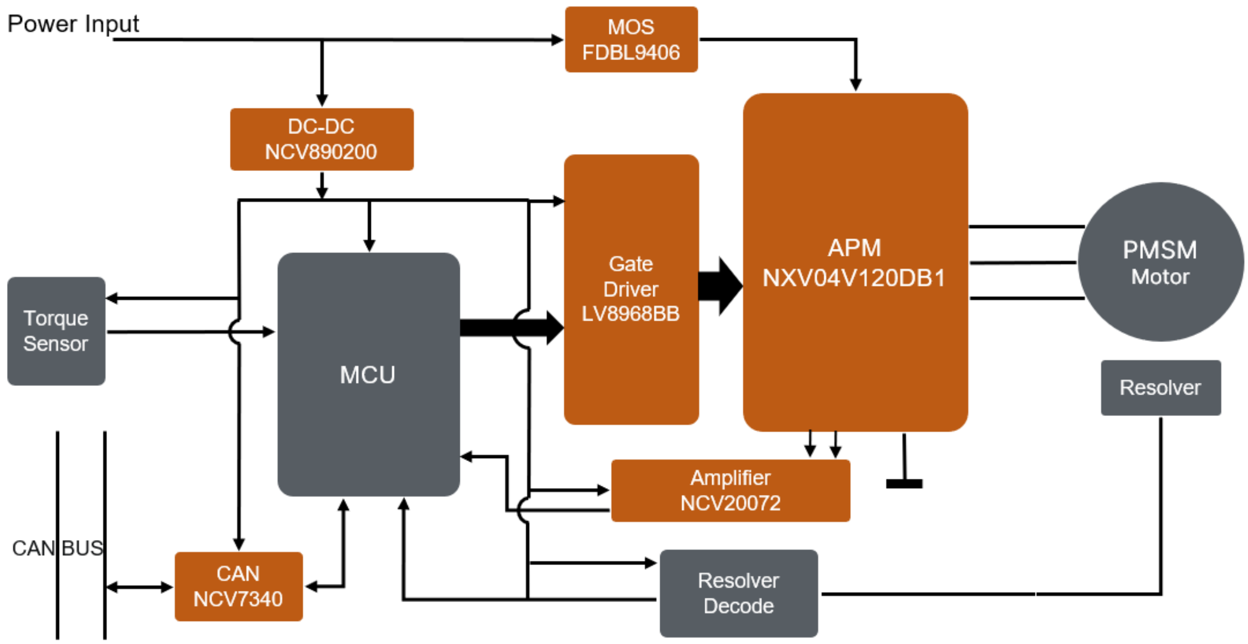

Figure 2. System Diagram of the EPS with APM19

The NXV04V120DB1 is a 40 V low Rds(on) automotive qualified power integrated module featuring a 3−phase MOSFET bridge optimized for an automotive 12 V motor inverter system. It includes a precision shunt resistor for current sensing, a NTC for temperature sensing, and a RC snubber circuit. The module utilizes **onsemi**'s trench MOSFET technology and it is designed to provide a very compact solution for the system. For more details on the NXV04V120DB1 please refer to the datasheet and the application notes on the NXV04V120DB1 web page.

The LV8968BB is a multi−purpose three−phase BLDC pre−driver for automotive applications and has been developed in compliance with ISO 26262. Six gate drivers provide 400 mA (typ) gate current to external power bridges allowing the use of low−resistance power FETs as well as logic level FETs. All FETs are protected against overcurrent, short−circuit, over−temperature and gate undervoltage. A multitude of protection and monitoring features make this device suitable for applications with functional safety requirements. Three independent low−side source pins allow multiple shunt measurements. The device also includes a programmable linear regulator, a fast current−sense amplifier, and a windowed watchdog for microcontroller support. The SPI interface allows for real time parameter setup and diagnostics. Critical system parameters can be programmed into non−volatile OTP memory. For more details on the LV8968BB please refer the datasheet and the application notes on the LV8968BB web page.

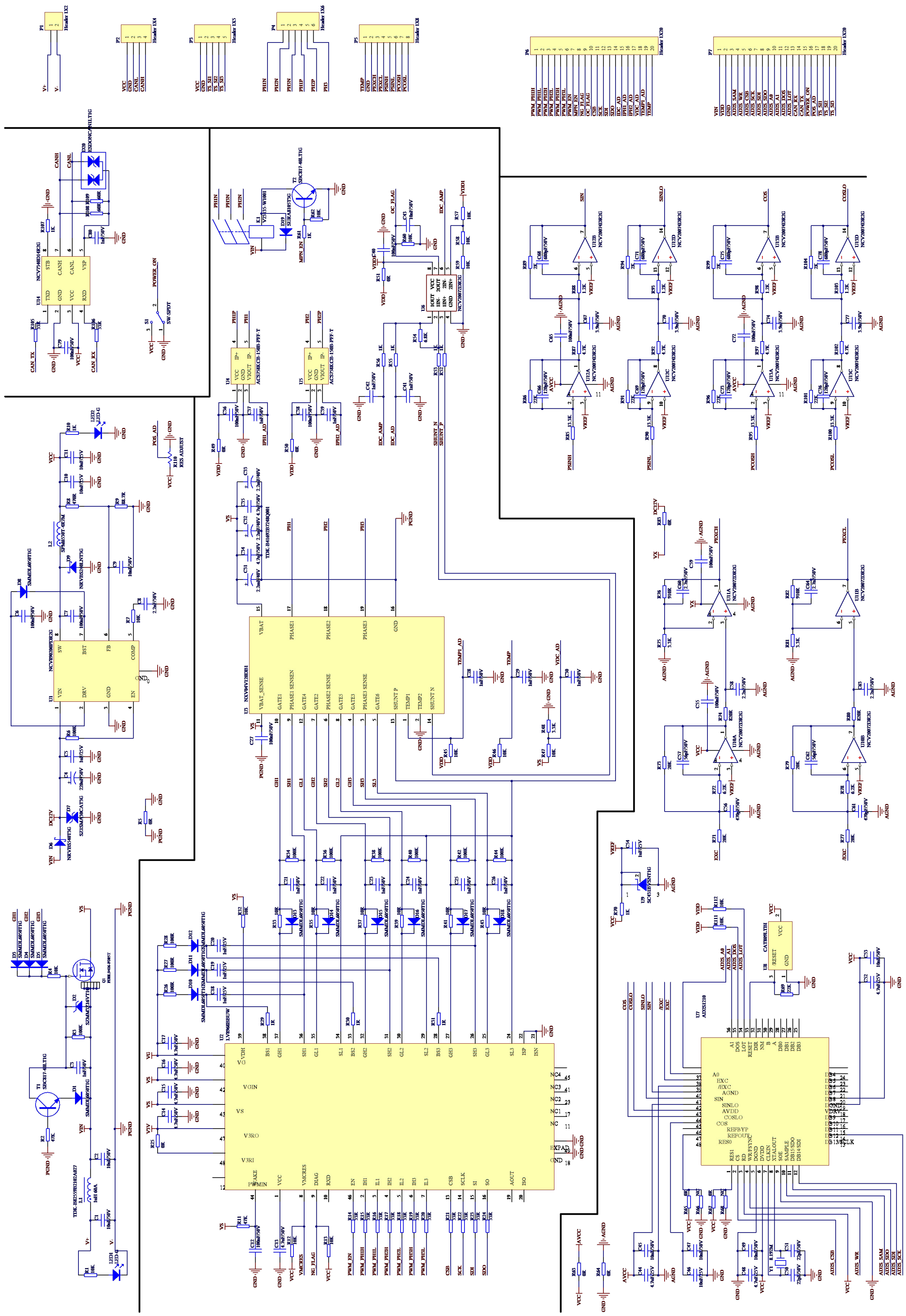

Figure 3. Schematic of the Main Board

The connector P1 connects 12 V main power to the power−board. Table 1 shows the signals.

Table 1. THE CONNECTION OF POWER INPUT

Pin No.	Direction	Description
1	V+	+12 V main power input
2	V−	GND

The connector P2 connects CAN communication to the power−board. Table 2 shows the signals.

Table 2. THE CONNECTION OF CAN COMMUNICATION

Pin No.	Direction	Description
1	VCC	+5 V
2	GND	GND
3	CANL	CAN communication
4	CANH	CAN communication

The connector P3 connects motor rotor signal (reserves) to the power−board. Table 3 shows the signals.

Table 3. THE CONNECTION OF POWER INPUT

Pin No.	Direction	Description
1	VCC	+5 V
2	GND	GND
3	TS−SI1	Channel 1 of rotor signal
4	TS−SI2	Channel 2 of rotor signal
5	TS−SI3	Channel 3 of rotor signal

The connector P4 connects the motor phase to the power−board. Table 4 shows the signals.

Table 4. THE CONNECTION OF MOTOR PHASES

Pin No.	Direction	Description
1	PH1N	The Phase1 Negative port
2	PH2N	The Phase2 Negative port
3	PH3N	The Phase3 Negative port
4	PH1P	The Phase1 Positive port
5	PH2P	The Phase2 Positive port
6	PH3P	The Phase3 Positive port

The connector P5 connects motor resolver−decode to the power−board. Table 5 shows the signals.

Table 5. THE CONNECTION OF MOTOR RESOLVER−DECODE

Pin No.	Direction	Description
1	TEMP	The temperature sensor positive port
2	GND	The temperature sensor negative port
3	PEXCH	The feedback of exciter positive port
4	PEXCL	The feedback of exciter negative port
5	PSINH	The Sin−Out of exciter positive port
6	PSINL	The Sin−Out of exciter negative port
7	PCOSH	The Cos−Out of exciter positive port
8	PCOSL	The Cos−Out of exciter negative port

The connector P6 connects control−signal to the MCU−board. Table 6 shows the signals.

Table 6. THE CONNECTION CONNECTS MCU TO MAIN BOARD

Pin No.	Direction	Description
1	PWM_PH1H	PWM1H to gate−driver
2	PWM_PH1L	PWM1L to gate−driver
3	PWM_PH2H	PWM2H to gate−driver
4	PWM_PH2L	PWM2L to gate−driver
5	PWM_PH3H	PWM3H to gate−driver
6	PWM_PH3L	PWM3L to gate−driver
7	PWM_EN	Enable signal of gate−driver
8	MPN_EN	Enable signal of relay
9	NG_FLAG	Fault signal from gate−driver
10	OC_FLAG	Over current signal from Power−Module
11	CSB	SPI CSB of connection to gate−driver
12	SCK	SPI SCK of connection to gate−driver
13	SDI	SPI SDI of connection to gate−driver
14	SDO	SPI SDO of connection to gate−driver
15	IDC_AD	Bus current sample signal
16	PH1_AD	Phase1 current sample signal
17	PH2_AD	Phase2 current sample signal
18	VDC_AD	Voltage sample signal
19	TEMP1_AD	The Temperature of Power−Module sample signal
20	TEMP_AD	The Temperature of resolver−decode sample signal

The connector P7 connects control−signal to the MCU−board. Table 7 shows the signals.

Table 7. THE CONNECTION CONNECTS MCU TO MAIN BOARD

Pin No.	Direction	Description
1	VIN	+12 V
2	VDD	+5 V for AD sample
3	GND	GND
4	AD2S_SAM	The SAM of AD2S1210
5	AD2S_WR	The WR of AD2S1210
6	AD2S_CSB	SPI CSB of connection to AD2S1210
7	AD2S_SCK	SPI SCK of connection to AD2S1210
8	AD2S_SDI	SPI SDI of connection to AD2S1210
9	AD2S_SDO	SPI SDO of connection to AD2S1210
10	AD2S_A0	The A0 of AD2S1210
11	AD2S_A1	The A1 of AD2S1210
12	AD2S_DOS	The DOS of AD2S1210
13	AD2S_LOT	The LOT of AD2S1210
14	CAN_RX	Connect CAN RXD of NCV7340 to MCU
15	CAN_TX	Connect CAN TXD of NCV7340 to MCU
16	POWER_ON	Connect switch on−off signal to MCU
17	POS_AD	Connect positioner voltage signal to MCU
18	TS_SI1	Connect channel1 of rotor signal to MCU
19	TS_SI2	Connect channel2 of rotor signal to MCU
20	TS_SI3	Connect channel3 of rotor signal to MCU

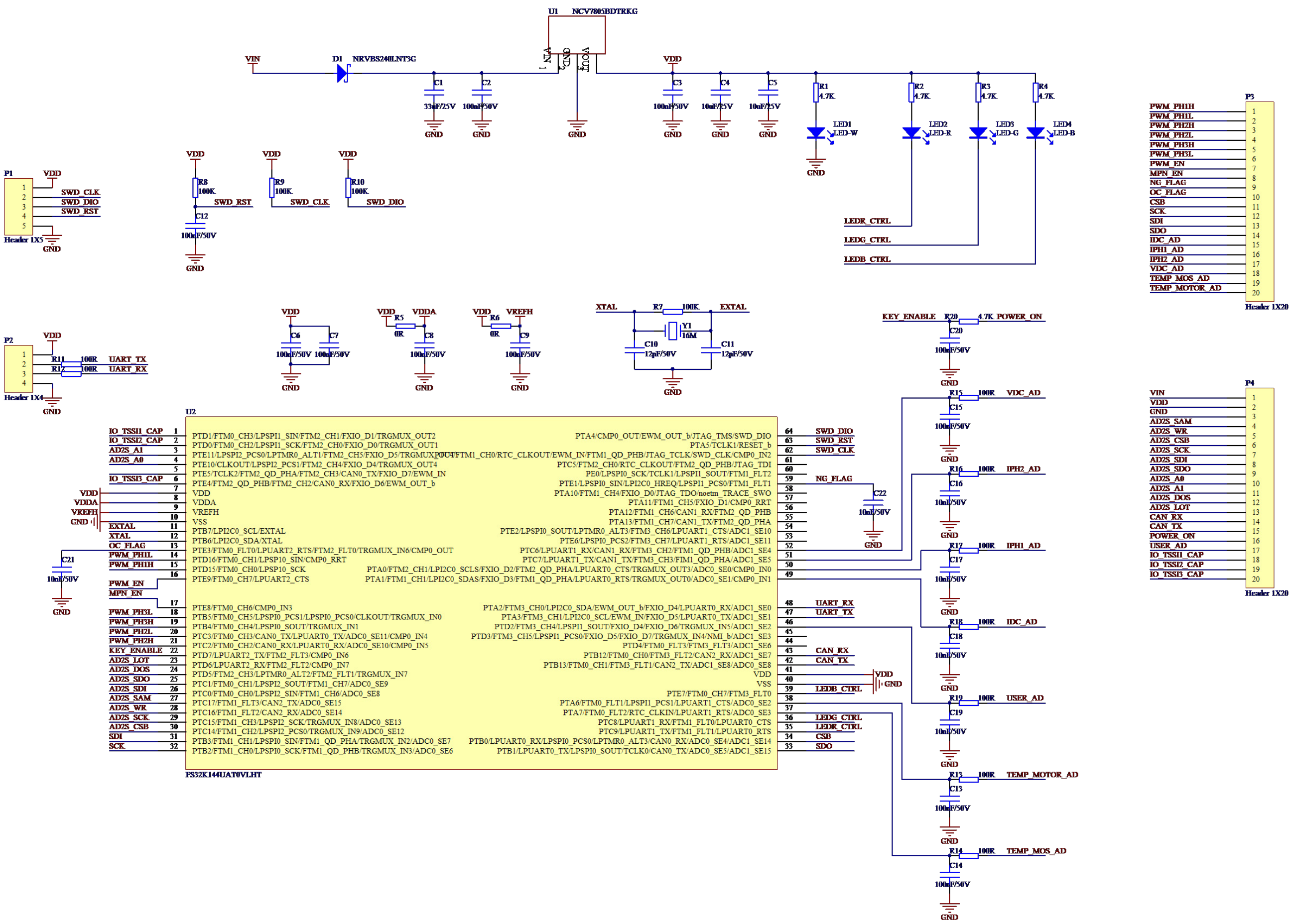

Figure 4. Schematic of the MCU Control Board

The connector P1 connects control−signal to the power−board. Table 8 shows the signals.

Table 8. THE CONNECTION CONNECTS MCU TO POWER−BOARD

Pin No.	Direction	Description
1	PWM_PH1H	PWM1H to gate−driver
2	PWM_PH1L	PWM1L to gate−driver
3	PWM_PH2H	PWM2H to gate−driver
4	PWM_PH2L	PWM2L to gate−driver
5	PWM_PH3H	PWM3H to gate−driver
6	PWM_PH3L	PWM3L to gate−driver
7	PWM_EN	Enable signal of gate−driver
8	MPN_EN	Enable signal of relay
9	NG_FLAG	Fault signal from gate−driver
10	OC_FLAG	Over current signal from Power−Module
11	CSB	SPI CSB of connection to gate−driver
12	SCK	SPI SCK of connection to gate−driver
13	SDI	SPI SDI of connection to gate−driver
14	SDO	SPI SDO of connection to gate−driver
15	IDC_AD	Bus current sample signal
16	PH1_AD	Phase1 current sample signal
17	PH2_AD	Phase2 current sample signal
18	VDC_AD	Voltage sample signal
19	TEMP1_AD	The Temperature of Power−Module sample signal
20	TEMP_AD	The Temperature of resolver−decode sample signal

The connector P2 connects control−signal to the power−board. The table 9 shows the signals.

Table 9. THE CONNECTION CONNECTS MCU TO POWER−BOARD

Pin No.	Direction	Description
1	VIN	+12 V
2	VDD	+5 V for AD sample
3	GND	GND
4	AD2S_SAM	The SAM of AD2S1210
5	AD2S_WR	The WR of AD2S1210
6	AD2S_CSB	SPI CSB of connection to AD2S1210
7	AD2S_SCK	SPI SCK of connection to AD2S1210
8	AD2S_SDI	SPI SDI of connection to AD2S1210
9	AD2S_SDO	SPI SDO of connection to AD2S1210
10	AD2S_A0	The A0 of AD2S1210
11	AD2S_A1	The A1 of AD2S1210
12	AD2S_DOS	The DOS of AD2S1210
13	AD2S_LOT	The LOT of AD2S1210
14	CAN_RX	Connect CAN RXD of NCV7340 to MCU
15	CAN_TX	Connect CAN TXD of NCV7340 to MCU
16	POWER_ON	Connect switch on−off signal to MCU
17	POS_AD	Connect positioner voltage signal to MCU
18	TS_SI1	Connect channel1 of rotor signal to MCU
19	TS_SI2	Connect channel2 of rotor signal to MCU
20	TS_SI3	Connect channel3 of rotor signal to MCU

SOFTWARE CODE OF FOC FOR MOTOR−DRIVER

```
void PhaseCurrent_Translate(void)
{
            tFloat Data_Iu,Data_Iv;
            Data_Iu = Ad_CurrentA - 0x7FF;
            if(Data_Iu >= 0x666)
                        Data_Iu = 0x666;
            if(Data_Iu <= -0x666)
                        Data_Iu = -0x666;
            Data_Iv = Ad_CurrentB - 0x7FF;
            if(Data_Iv >= 0x666)
                        Data_Iv = 0x666;
            if(Data_Iv <= -0x666)
                        Data_Iv = -0x666;
            FOC.Iu = MLIB_Mul((MLIB_Div(Data_Iu, (tFloat)0x666)), _de_CurrentSensor_IMax);
            FOC.Iv = MLIB_Mul((MLIB_Div(Data_Iv, (tFloat)0x666)), _de_CurrentSensor_IMax);
            FOC.Iw = MLIB_Neg_FLT(MLIB_Add(FOC.Iu,FOC.Iv));
}
void RotorAngle_Translate(void)
{
            volatile unsigned long tm_Long;
#if         _de_Use_HallThree
            HallThree.Angle_Elec = HallThree_Angle();
            Rotor.Angle_ElecTheta = (tU16)((tU32)(HallThree.Angle_Elec) * 360 / 65536);
#endif
#if         _de_Use_Resolver
            Resolver.Angle_Elec = AD2S1210_Angle();
            Rotor.Angle_ElecTheta = (tU16)((tU32)(Resolver.Angle_Elec) * 360 / 65536);
#endif
#if         _de_Use_SoftwareFix
            SoftwareFix.Angle_Elec += 24;
            Rotor.Angle_ElecTheta = (tU16)((tU32)(SoftwareFix.Angle_Elec) * 360 / 65536);
#endif
            FOC.CosTheta = MLIB_Div((tFloat)(CosTable[Rotor.Angle_ElecTheta]), (tFloat)(32767));
            FOC.SinTheta = MLIB_Div((tFloat)(SinTable[Rotor.Angle_ElecTheta]), (tFloat)(32767));
}
void RotorSpeed_Translate(void)
{
#if         _de_Use_HallThree
            HallThree.Speed_Elec = HallThree_Speed();
            HallThree.Speed_RPM = (HallThree.Speed_Elec)/(HallThree.Pola_Num);
            Rotor.Speed_RPM = HallThree.Speed_RPM;
#endif
#if         _de_Use_Resolver
            Resolver.Speed_Elec = AD2S1210_Speed();
            Resolver.Speed_RPM = (Resolver.Speed_Elec)/(Resolver.Pola_Num);
            Rotor.Speed_RPM = Resolver.Speed_RPM;
#endif
#if         _de_Use_SoftwareFix
            SoftwareFix.Speed_Elec = 24*50;
            SoftwareFix.Speed_RPM = (SoftwareFix.Speed_Elec)/(SoftwareFix.Pola_Num);
            Rotor.Speed_RPM = Resolver.Speed_RPM;
#endif
}
void CLARK_Translate(void)
{
//          Ialpha = Iu;
//          Ibeta = (Iu + Iv*2)*SQRT3INT
            FOC.Ialpha = FOC.Iu;
            FOC.Ibeta = MLIB_Mul(MLIB_Add(FOC.Iu, MLIB_Mul(FOC.Iu, 2.0F)), _de_SQRT3INT_Float);
}
void PARK_Translate(void)
{
//          Id = Ialpha*AngalCos + Ibeta*AngalSin;
//          Iq = Ibeta*AngalCos - Ialpha*AngalSin;
            FOC.Id = MLIB_Add(MLIB_Mul(FOC.Ialpha, FOC.CosTheta), MLIB_Mul(FOC.Ibeta,
FOC.SinTheta));
```

```
            FOC.Iq = MLIB_Sub(MLIB_Mul(FOC.Ibeta, FOC.CosTheta), MLIB_Mul(FOC.Ialpha, FOC.SinTheta
));
}
void IPARK_Translate(void)
{
//          Ualpha = Ud*AngalCos - Uq*AngalSin;
//          Ubeta = Ud*AngalSin + Uq*AngalCos;
            FOC.Ualpha = MLIB_Sub(MLIB_Mul(FOC.Ud, FOC.CosTheta), MLIB_Mul(FOC.Uq, FOC.SinTheta));
            FOC.Ubeta = MLIB_Add(MLIB_Mul(FOC.Ud, FOC.SinTheta), MLIB_Mul(FOC.Uq, FOC.CosTheta));
}
void SVW_Translate(void)
{
            tFloat Data1,Data2,Data3;
            tU8 Num1,Num2,Num3,Num;

            SVW.Ualpha = FOC.Ualpha;
            SVW.Ubeta = FOC.Ubeta;

//          Data1 = SVW.Ubeta;
//          Data2 = -_IQdiv2(SVW.Ubeta) + _IQmpy(SVW.Ualpha, _IQ15toIQ(SQRT3DIV2));
//          Data3 = -_IQdiv2(SVW.Ubeta) -_IQmpy(SVW.Ualpha, _IQ15toIQ(SQRT3DIV2));
            Data1 = SVW.Ubeta;
            Data2 = MLIB_Add(MLIB_Neg_FLT(MLIB_Div(SVW.Ubeta, 2.0F)), MLIB_Mul(SVW.Ualpha,
_de_SQRT3DIV2_Float));
            Data3 = MLIB_Sub(MLIB_Neg_FLT(MLIB_Div(SVW.Ubeta, 2.0F)), MLIB_Mul(SVW.Ualpha,
_de_SQRT3DIV2_Float));

            Num1 = (Data1 >= 0) ? 1 : 0;
            Num2 = (Data2 >= 0) ? 1 : 0;
            Num3 = (Data3 >= 0) ? 1 : 0;
            Num = Num1 + Num2*2 + Num3*4;
            if(Num == 1)
                        SVW.Sector = 2;
            else if(Num == 2)
                        SVW.Sector = 6;
            else if(Num == 3)
                        SVW.Sector = 1;
            else if(Num == 4)
                        SVW.Sector = 4;
            else if(Num == 5)
                        SVW.Sector = 3;
            else
                        SVW.Sector = 5;

//          SVW.Tx = SVW.Ubeta;
//          SVW.Ty = _IQdiv2(SVW.Ubeta) + _IQmpy(_IQ15toIQ(SQRT3DIV2), SVW.Ualpha);
//          SVW.Tz = -_IQdiv2(SVW.Ubeta) + _IQmpy(_IQ15toIQ(SQRT3DIV2), SVW.Ualpha);
            SVW.Tx = SVW.Ubeta;
            SVW.Ty = MLIB_Add(MLIB_Div(SVW.Ubeta, 2.0F), MLIB_Mul(SVW.Ualpha,
_de_SQRT3DIV2_Float));
            SVW.Tz = MLIB_Add(MLIB_Neg_FLT(MLIB_Div(SVW.Ubeta, 2.0F)), MLIB_Mul(SVW.Ualpha,
_de_SQRT3DIV2_Float));
            switch(SVW.Sector)
            {
                        case 1:
                          SVW.Tk = (tU16)(MLIB_Mul(SVW.Tz, SVW.Ts));
                          SVW.Tk1 = (tU16)(MLIB_Mul(SVW.Tx, SVW.Ts));
                          if((SVW.Tk + SVW.Tk1) > SVW.Ts)
                          {
                              SVW.Tk = SVW.Tk*SVW.Ts/(SVW.Tk+SVW.Tk1);
                              SVW.Tk1 = SVW.Tk1*SVW.Ts/(SVW.Tk+SVW.Tk1);
                          }
                          SVW.T0 = SVW.Ts - SVW.Tk - SVW.Tk1;
                          SVW.Tw = SVW.T0/2;
                          SVW.Tv = SVW.T0/2 + SVW.Tk1;
                          SVW.Tu = SVW.T0/2 + SVW.Tk1 + SVW.Tk;
                          break;
                        case 2:
```

```
                    SVW.Tk = (tU16)(MLIB_Mul(SVW.Ty, SVW.Ts));
                    SVW.Tk1 = (tU16)(MLIB_Mul(-SVW.Tz, SVW.Ts));
                    if((SVW.Tk + SVW.Tk1) > SVW.Ts)
                    {
                        SVW.Tk = SVW.Tk*SVW.Ts/(SVW.Tk+SVW.Tk1);
                        SVW.Tk1 = SVW.Tk1*SVW.Ts/(SVW.Tk+SVW.Tk1);
                    }
                    SVW.T0 = SVW.Ts - SVW.Tk - SVW.Tk1;
                    SVW.Tw = SVW.T0/2;
                    SVW.Tu = SVW.T0/2 + SVW.Tk;
                    SVW.Tv = SVW.T0/2 + SVW.Tk + SVW.Tk1;
                    break;
                case 3:
                    SVW.Tk = (tU16)(MLIB_Mul(SVW.Tx, SVW.Ts));
                    SVW.Tk1 = (tU16)(MLIB_Mul(-SVW.Ty, SVW.Ts));
                    if((SVW.Tk + SVW.Tk1) > SVW.Ts)
                    {
                        SVW.Tk = SVW.Tk*SVW.Ts/(SVW.Tk+SVW.Tk1);
                        SVW.Tk1 = SVW.Tk1*SVW.Ts/(SVW.Tk+SVW.Tk1);
                    }
                    SVW.T0 = SVW.Ts - SVW.Tk - SVW.Tk1;
                    SVW.Tu = SVW.T0/2;
                    SVW.Tw = SVW.T0/2 + SVW.Tk1;
                    SVW.Tv = SVW.T0/2 + SVW.Tk1 + SVW.Tk;
                    break;
                case 4:
                    SVW.Tk = (tU16)(MLIB_Mul(-SVW.Tz, SVW.Ts));
                    SVW.Tk1 = (tU16)(MLIB_Mul(-SVW.Tx, SVW.Ts));
                    if((SVW.Tk + SVW.Tk1) > SVW.Ts)
                    {
                        SVW.Tk = SVW.Tk*SVW.Ts/(SVW.Tk+SVW.Tk1);
                        SVW.Tk1 = SVW.Tk1*SVW.Ts/(SVW.Tk+SVW.Tk1);
                    }
                    SVW.T0 = SVW.Ts - SVW.Tk - SVW.Tk1;
                    SVW.Tu = SVW.T0/2;
                    SVW.Tv = SVW.T0/2 + SVW.Tk;
                    SVW.Tw = SVW.T0/2 + SVW.Tk + SVW.Tk1;
                    break;
                case 5:
                    SVW.Tk = (tU16)(MLIB_Mul(-SVW.Ty, SVW.Ts));
                    SVW.Tk1 = (tU16)(MLIB_Mul(SVW.Tz, SVW.Ts));
                    if((SVW.Tk + SVW.Tk1) > SVW.Ts)
                    {
                        SVW.Tk = SVW.Tk*SVW.Ts/(SVW.Tk+SVW.Tk1);
                        SVW.Tk1 = SVW.Tk1*SVW.Ts/(SVW.Tk+SVW.Tk1);
                    }
                    SVW.T0 = SVW.Ts - SVW.Tk - SVW.Tk1;
                    SVW.Tv = SVW.T0/2;
                    SVW.Tu = SVW.T0/2 + SVW.Tk1;
                    SVW.Tw = SVW.T0/2 + SVW.Tk1 + SVW.Tk;
                    break;
                default:
                    SVW.Tk = (tU16)(MLIB_Mul(-SVW.Tx, SVW.Ts));
                    SVW.Tk1 = (tU16)(MLIB_Mul(SVW.Ty, SVW.Ts));
                    if((SVW.Tk + SVW.Tk1) > SVW.Ts)
                    {
                        SVW.Tk = SVW.Tk*SVW.Ts/(SVW.Tk+SVW.Tk1);
                        SVW.Tk1 = SVW.Tk1*SVW.Ts/(SVW.Tk+SVW.Tk1);
                    }
                    SVW.T0 = SVW.Ts - SVW.Tk - SVW.Tk1;
                    SVW.Tv = SVW.T0/2;
                    SVW.Tw = SVW.T0/2 + SVW.Tk;
                    SVW.Tu = SVW.T0/2 + SVW.Tk + SVW.Tk1;
                    break;
            }
}
void FOC_Control(void)
{
```

```
            static volatile tU8 st_SpeedPI_Count = 0;
            static volatile tU8 st_CurrentPI_Count = 0;
            volatile tFloat tm_Rate;
            if(Ad_Knob >= 0x8FF)
            {
                        if(Ad_Knob >= 0xEFF)
                          tm_Rate = (1.0F);
                        else
                          tm_Rate = MLIB_Div((Ad_Knob - 0x8FF), (0xEFF - 0x8FF));
            }
            else if(Ad_Knob <= 0x6FF)
            {
                        if(Ad_Knob <= 0x0FF)
                          tm_Rate = (-1.0F);
                        else
                          tm_Rate= MLIB_Neg_FLT(MLIB_Div((0x6FF - Ad_Knob), (0x6FF - 0x0FF)));
            }
            else
            {
                        tm_Rate = (0.0F);
            }
            Target.Speed = MLIB_Mul(_de_TargetSpeed_MaxData, tm_Rate);
            Target.Current = MLIB_Mul(_de_TargetCurrent_MaxData, tm_Rate);
            Target.Modulation = MLIB_Mul(_de_TargetModulation_MaxData, tm_Rate);
            PhaseCurrent_Translate();
            RotorAngle_Translate();
            CLARK_Translate();
            PARK_Translate();

            if(++st_SpeedPI_Count >= 10)
            {
                        // 100ms*10 = 2ms
                        st_SpeedPI_Count = 0;
                        RotorSpeed_Translate();
                        PI_Speed.SetValue = Target.Speed;
                        PI_Speed.FeedBack = Rotor.Speed_RPM;
                        PI_Control(&PI_Speed);
//                      PI_Speed.OutF    =    _IQ15mpy(_de_PISpeedFilter_K1,PI_Speed.OutF)     +
_IQ15mpy(_de_PISpeedFilter_K2,PI_Speed.Out);
                        PI_Speed.OutF = PI_Speed.Out;
            }
#if         _de_Use_SpeedLoop
            PI_Id.SetValue = (0.0F);
            PI_Iq.SetValue = PI_Speed.OutF;
#else
            PI_Id.SetValue = (0.0F);
            PI_Iq.SetValue = Target.Current;
#endif

            if(++st_CurrentPI_Count >= 2)
            {
                        st_CurrentPI_Count = 0;
                        PI_Id.FeedBack = FOC.Id;
                        PI_Iq.FeedBack = FOC.Iq;
                        PI_Control(&PI_Id);
//                      PI_Id.OutF = _IQ15mpy(_de_PICurrentFilter_K1,PI_Id.OutF) +
_IQ15mpy(_de_PICurrentFilter_K2,PI_Id.Out);
                        PI_Id.OutF = PI_Id.Out;
                        PI_Control(&PI_Iq);
//                      PI_Iq.OutF = _IQ15mpy(_de_PICurrentFilter_K1,PI_Iq.OutF)            +
_IQ15mpy(_de_PICurrentFilter_K2,PI_Iq.Out);
                        PI_Iq.OutF = PI_Iq.Out;
            }
#if         _de_Use_CurrentLoop
            FOC.Ud = PI_Id.OutF;
            FOC.Uq = PI_Iq.OutF;
#else
            FOC.Ud = (0.0F);
```

```
//          FOC.Uq = Target.Modulation;
            if(FOC.Uq < Target.Modulation)
            {
                        FOC.Uq += (0.0001F);
                        if(FOC.Uq > Target.Modulation)
                           FOC.Uq = Target.Modulation;
            }
            else
            {
                        FOC.Uq -= (0.0001F);
                        if(FOC.Uq < Target.Modulation)
                           FOC.Uq = Target.Modulation;
            }
#endif

            IPARK_Translate();
            SVW_Translate();
}
```

TEST WAVEFORM

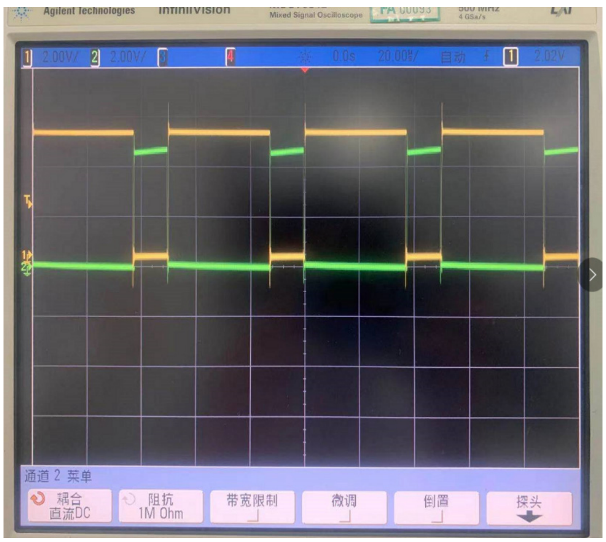

Figure 5. Waveform of Complementary PWM

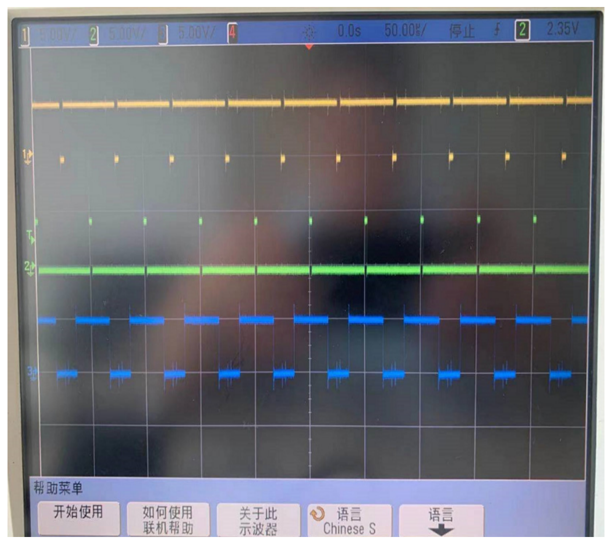

Figure 6. 3 Phase PWM for FOC

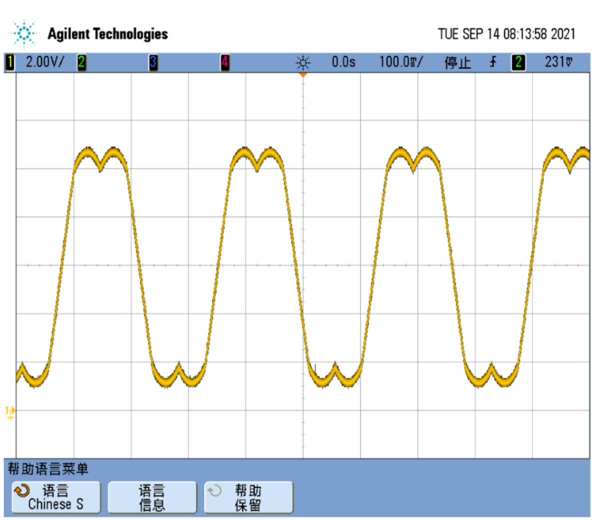

Figure 7. Waveform of Phase−Voltage

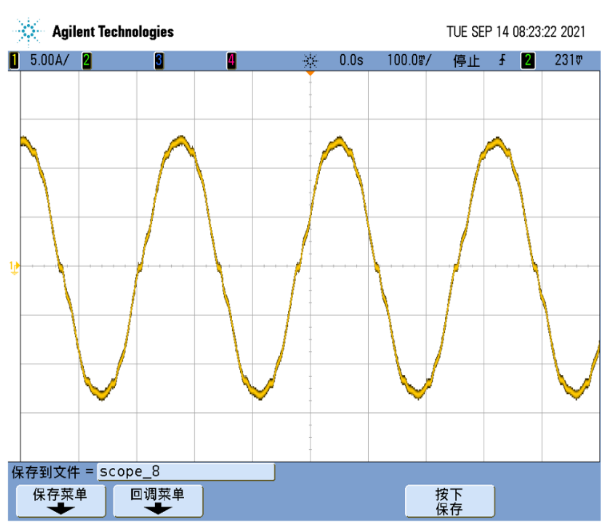

Figure 8. Waveform of Phase−Current

PCB LAYOUT

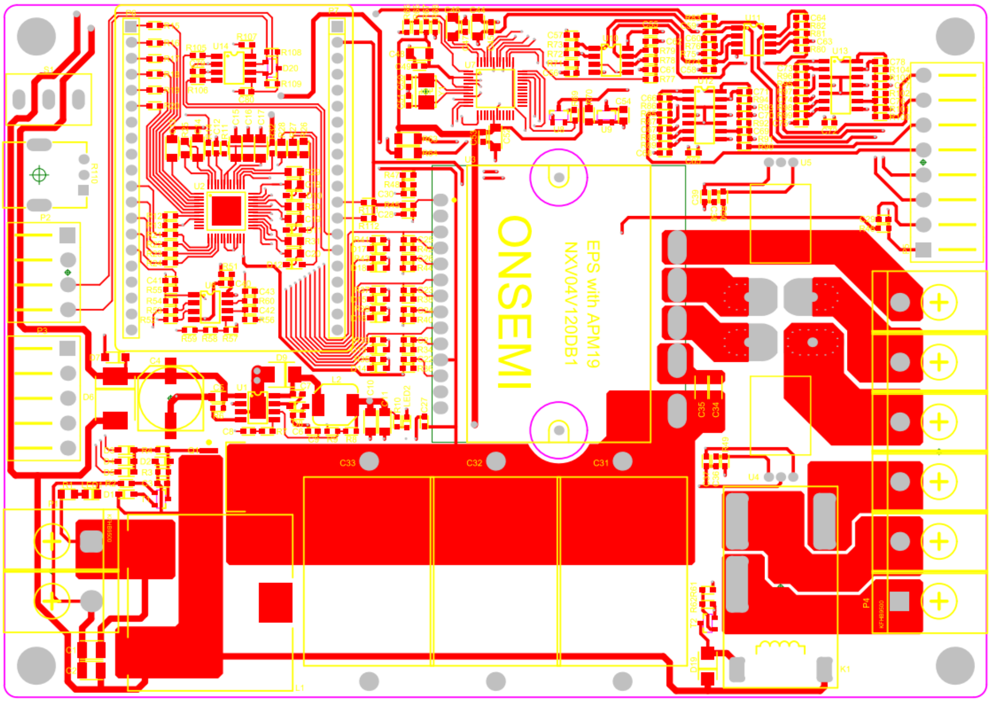

Figure 9. Top Side View of Power−Board (155 x 110 x 2 mm. 2oz)

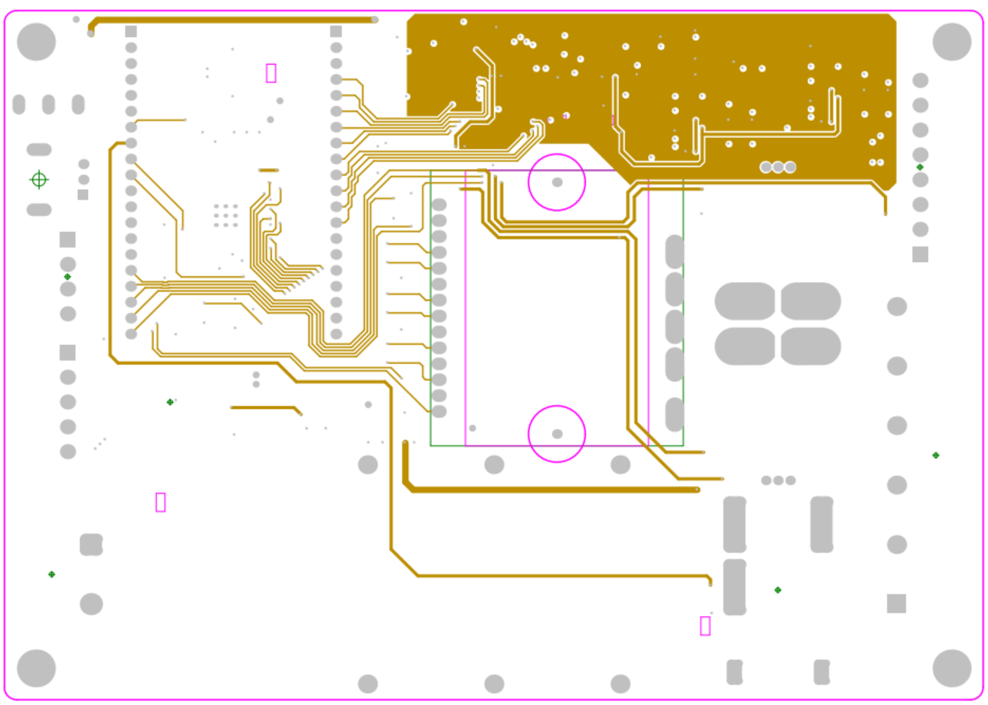

Figure 10. Middle 1 Side View of Power−Board (155 x 110 x 2 mm. 1oz)

PCB LAYOUT (continued)

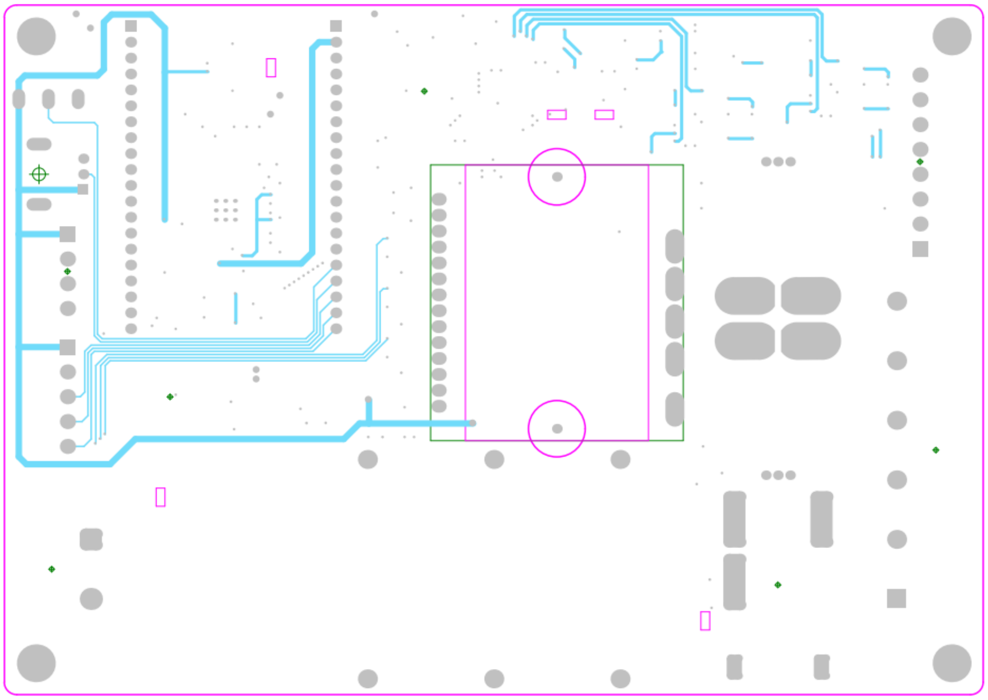

Figure 11. Middle 2 Side View of Power−Board (155 x 110 x 2 mm. 1oz)

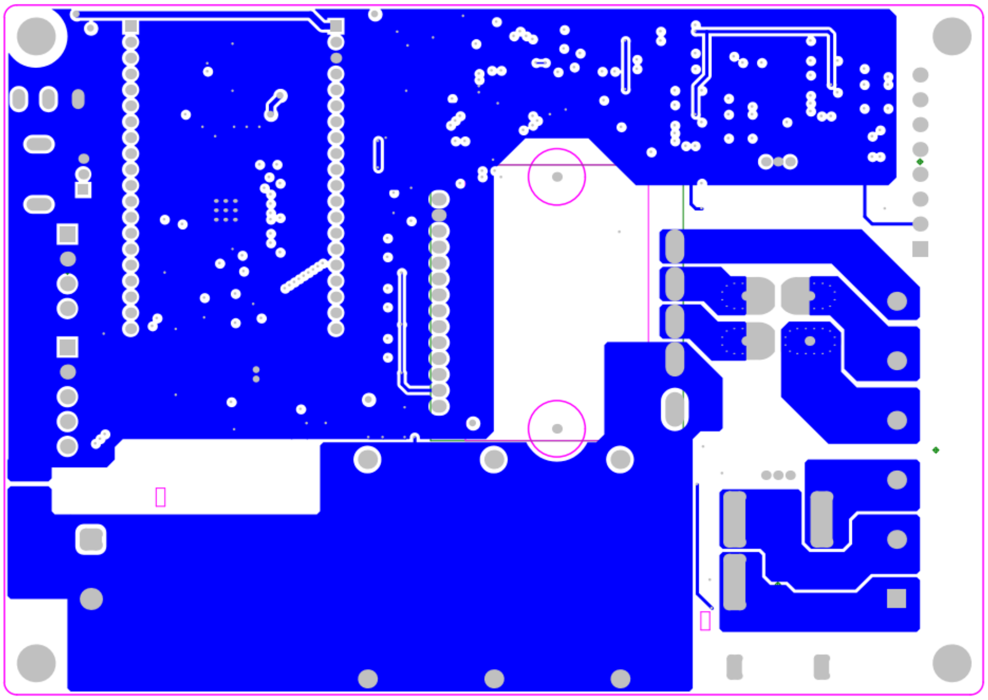

Figure 12. Bottom Side View of Power−Board (155 x 110 x 2 mm. 2oz)

PCB LAYOUT (continued)

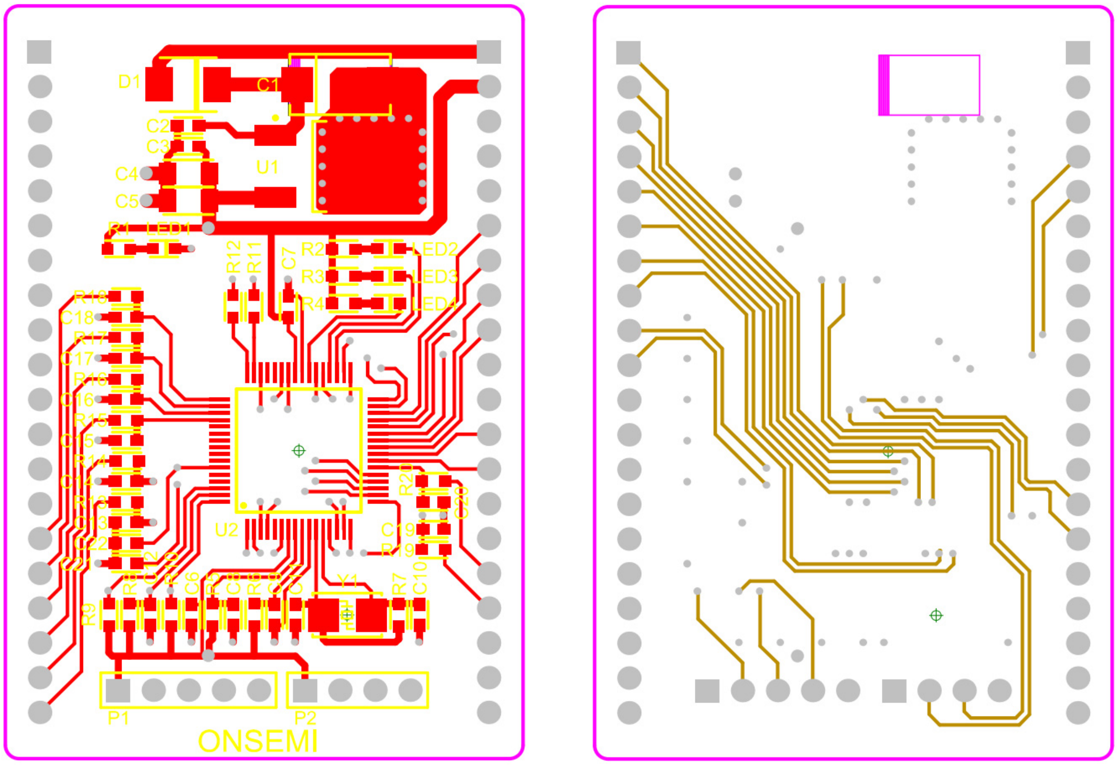

Figure 13. Top Side and Middle Side View of MCU−Board (55 x 37.5 x 2 mm. 1oz)

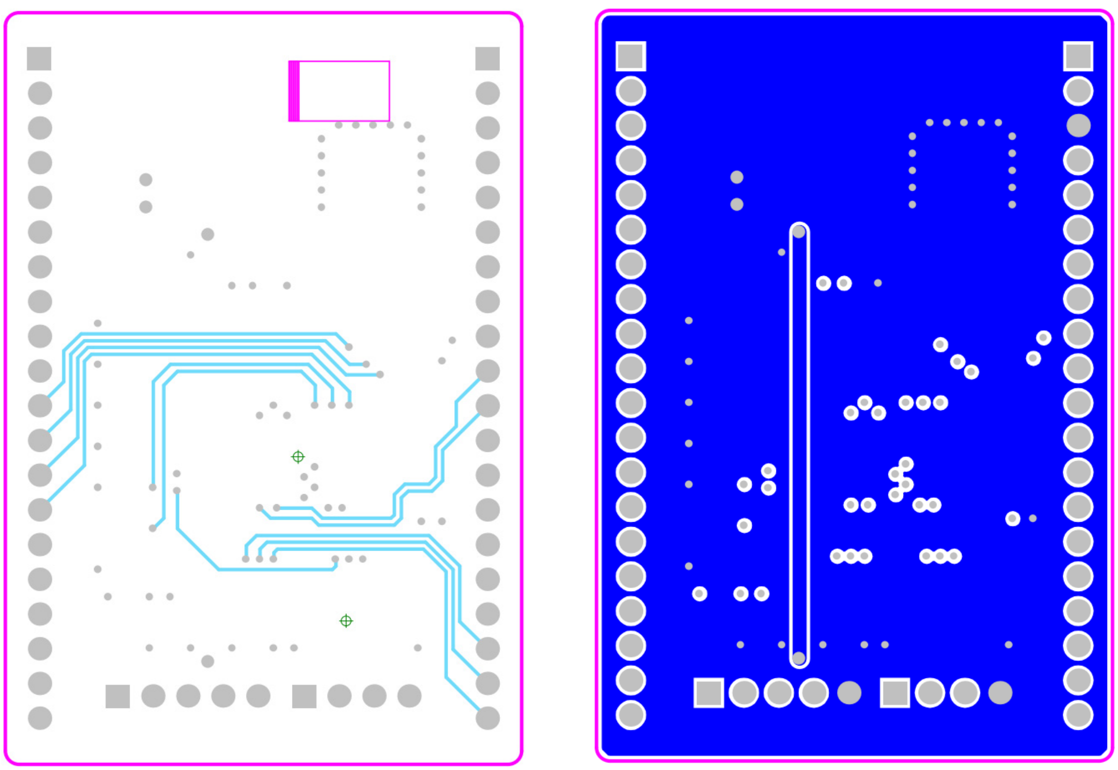

Figure 14. Middle 2 Side and Bottom Side View of MCU−Board (55 x 37.5 x 2 mm. 1oz)

BILL OF MATERIALS

Table 10. BOM LIST OF MCU−BOARD

Manufacturer Part Number	Manufacturer	Description	Designator	Qty
TAJD336K025RNJ	AVX	Tantalum Capacitors, CASE−D_7343, 33 μF/25 V	C1	1
GCM1885G1H104FA16D	MURATA	MLCC, 0603, 100 nF/50 V	C2, C3, C6, C7, C8, C9, C12, C13, C14, C15, C20	11
GCM31CC71E106KA03K	MURATA	MLCC, 1206, 10 μF/25 V	C4, C5	2
GCM1885G1H120FA16D	MURATA	MLCC, 0603, 12 pF/50 V	C10, C11	2
GCM1885G1H103FA16D	MURATA	MLCC, 0603, 10 nF/50 V	C16, C17, C18, C19, C21, C22	6
NRVBS240LNT3G	**onsemi**	Schottky Diode, SMB	D1	1
19−217UWD/S365−2/TR8	EVERLIGHT	19−217 SMD LED, 0603, Write	LED1	1
19−21/RC6−FP1Q2L/3T	EVERLIGHT	19−21 SMD LED, 0603, Red	LED2	1
19−217/GHC−YR1S2/3T	EVERLIGHT	19−217 SMD LED, 0603, Green	LED3	1
19−21/Y2C−CP1Q2B/3T	EVERLIGHT	19−21 SMD LED, 0603, Yellow	LED4	1
A2541WV−5P	CJT	2.54 mm Pitch Connector, 5P	P1	1
A2541WV−4P	CJT	2.54 mm Pitch Connector, 4P	P2	1
A2541WV−20P	CJT	2.54 mm Pitch Connector, 20P	P3, P4	2
CRCW06034K70FKEA	VISHAY	Resistance, 0603, 4.7K/1%	R1, R2, R3, R4, R20	5
CRCW06030000Z0EA	VISHAY	Resistance, 0603, 0R/1%	R5, R6	2
CRCW0603100KFKEA	VISHAY	Resistance, 0603, 100K/1%	R7, R8, R9, R10	4
CRCW0603100RFKEA	VISHAY	Resistance, 0603, 100R/1%	R11, R12, R13, R14, R15, R16, R17, R18, R19	9
NCV7805BDTRKG	**onsemi**	Positive Voltage Regulators, 5 V/1 A	U1	1
FS32K144UAT0VLHT	NXP	MCU, Arm Cortex−M4F/M0+, LQFP64−10x10	U2	1
NX5320GA 16 MHz	NDK	Crystal, 5320, 16 MHz	Y1	1

Table 11. BOM LIST OF POWER−BOARD

Manufacturer Part Number	Manufacturer	Description	Designator	Qty
GCM32EC71H106KA03K	MURATA	MLCC, 1210, 10 μF/50 V	C1, C2	2
GCM1885G1H102FA16D	MURATA	MLCC, 0603, 1 nF/50 V	C3, C21, C22, C23, C24, C25, C26, C28, C29, C30, C37, C39, C41, C42, C80	15
EEEFN1H221V	PANASONIC	Capacitor, 220 μF/50 V	C4	1
GCM21BR71E105KA56L	MURATA	MLCC, 0805, 1 μF/25 V	C5, C18, C19, C20, C54	5
GCM1885G1H104FA16D	MURATA	MLCC, 0603, 100 nF/50 V	C6, C7, C12, C27, C36, C38, C40, C55, C59, C65, C72, C79	12
GCM1885G1H222FA16D	MURATA	MLCC, 0603, 2.2 nF/50 V	C8, C58, C63	3
GCM1885G1H103FA16D	MURATA	MLCC, 0603, 10 nF/50 V	C9, C43, C45, C47, C49, C53	6
GCM31CC71E106KA03K	MURATA	MLCC, 1206, 10 μF/50 V	C10, C11, C46	3
GCM31CC71H475KA03K	MURATA	MLCC, 1206, 4.7 μF/50 V	C13, C14, C15, C16, C17, C34, C35	7
B41692C7248Q001	TDK	Capacitor, 2.2 mF/40 V	C31, C32, C33	3
GCM31CR71E475KA55L	MURATA	MLCC, 1206, 4.7 μF/25 V	C44, C48, C52	3
GCM1885G1H220FA16D	MURATA	MLCC, 0603, 22 pF/50 V	C50, C51	2
GCM1885G1H471FA16D	MURATA	MLCC, 0603, 470 pF/50 V	C56, C61	2
GCM1885G1H560FA16D	MURATA	MLCC, 0603, 56 pF/50 V	C57, C62	2

Table 11. BOM LIST OF POWER−BOARD (continued)

Manufacturer Part Number	Manufacturer	Description	Designator	Qty
GCM1885G1H272FA16D	MURATA	MLCC, 0603, 2.7 nF/50 V	C60, C64	2
GCM1885G1H121FA16D	MURATA	MLCC, 0603, 120 pF/50 V	C66, C69, C73, C76	4
GCM1885G1H392FA16D	MURATA	MLCC, 0603, 3.9 nF/50 V	C67, C70, C74, C77	4
GCM1885G1H681FA16D	MURATA	MLCC, 0603, 680 pF/50 V	C68, C71, C75, C78	4
SZMM3Z16VT1G	**onsemi**	Zener Diode, 16 V, SOD−323	D2	1
SMMDL6050T1G	**onsemi**	Switch Diode, 70 V, SOD−323	D1, D3, D4, D5, D8, D10, D11, D12, D13, D14, D15, D16, D17, D18	14
NRVBS540T3G	**onsemi**	Schottky Diode, 40 V/5 A, SMC	D6	1
SZ1SMA30CAT3G	**onsemi**	TVS Bipolar Diode, SOD−123	D7	1
NRVBS240LNT3G	**onsemi**	Schottky Diode, 40 V/2 A, SMB	D9	1
SURA8105T3G	**onsemi**	Switch Diode, Ultra−Fast Recovery, 50V/1A, SMA	D19	1
ESDONCAN1LT1G	**onsemi**	CAN Bus Protrctor, SOT−23	D20	1
V23135−W1001	Tyco Elctronics	Star Point Relay SPR, 90 A, 32 x 17.5 x 18	K1	1
B82559B−A027	TDK	Power Inductor, 1 μH/60 A	L1	1
SPM6530T−4R7M	TDK	Power Inductor, 4.7 μH/5.6 A	L2	1
19−217/GHC−YR1S2/3T	EVERLIGHT	19−217 SMD LED, 0603, Green	LED1	1
19−217/GHC−YR1S2/3T	EVERLIGHT	19−217 SMD LED, 0603, Green	LED2	1
HB9500−9.5−2P	KANGNEX	9.5 mm Connector, 2P	P1	1
HT396R−3.96−4P	KANGNEX	3.96 mm Connector, 4P	P2	1
HT396R−3.96−5P	KANGNEX	3.96 mm Connector, 5P	P3	1
HB9500−9.5−6P	KANGNEX	9.5 mm Connector, 6P	P4	1
HT396R−3.96−8P	KANGNEX	3.96 mm Connector, 8P	P5	1
A2541HWV−20P	CJT	2.54 mm Pitch Connector, 20P	P6, P7	2
FDBL9406−F085T	**onsemi**	N−Channel MOSFET, 40 V, 240 A, H−PSOF8L	Q1	1
CRCW080510K0FKEA	VISHAY	Resistance, 0805, 10K/1%	R1	1
CRCW060347K0FKEA	VISHAY	Resistance, 0603, 47K/1%	R2, R11	2
CRCW0603100KFKEA	VISHAY	Resistance, 0603, 100K/1%	R3, R6, R34, R36, R38, R40, R42, R44	8
CRCW060310K0FKEA	VISHAY	Resistance, 0603, 10K/1%	R4, R7, R12, R13, R32, R45, R46, R47, R57, R58, R59, R60, R62, R111, R112	15
CRCW12060000Z0EA	VISHAY	Resistance, 1206, 0R/1%	R5, R63, R64	3
CRCW0603470RFKEA	VISHAY	Resistance, 0603, 470R/1%	R8	1
CRCW060388R7FKEA	VISHAY	Resistance, 0603, 88.7R/1%	R9	1
CRCW06031K00FKEA	VISHAY	Resistance, 0603, 1K/1%	R10, R52, R53, R55, R56, R61, R107	7
CRCW060333R0FKEA	VISHAY	Resistance, 0603, 33R/1%	R14, R15, R16, R17, R18, R19, R20, R21, R22, R23, R24, R105, R106	13
CRCW08050000Z0EA	VISHAY	Resistance, 0805, 0R/1%	R25	1
CRCW0805100RFKEA	VISHAY	Resistance, 0805, 100R/1%	R26, R27, R28	3
CRCW08051R00FKEA	VISHAY	Resistance, 0805, 1R/1%	R29, R30, R31	3
CRCW060310R0FKEA	VISHAY	Resistance, 0603, 10R/1%	R33, R35, R37, R39, R41, R43	6
CRCW06033K30FKEA	VISHAY	Resistance, 0603, 3.3K/1%	R48, R75, R81	3
CRCW06030000Z0EA	VISHAY	Resistance, 0603, 0R/1%	R49, R50, R51, R65, R67, R83	6
CRCW06036K80FKEA	VISHAY	Resistance, 0603, 6.8K/1%	R54	1

Table 11. BOM LIST OF POWER−BOARD (continued)

Manufacturer Part Number	Manufacturer	Description	Designator	Qty
NC		Resistance	R66, R68	2
CRCW060322K0FKEA	VISHAY	Resistance, 0603, 22K/1%	R69, R86, R91, R96, R101	5
CRCW08051K00FKEA	VISHAY	Resistance, 0805, 1K/1%	R70	1
CRCW060320K0FKEA	VISHAY	Resistance, 0603, 20K/1%	R71, R73, R77, R79	4
CRCW06036K20FKEA	VISHAY	Resistance, 0603, 6.2K/1%	R72, R78	2
CRCW0603820RFKEA	VISHAY	Resistance, 0603, 820R/1%	R74, R80	2
CRCW0603910RFKEA	VISHAY	Resistance, 0603, 910R/1%	R76, R82	2
CRCW060313K3FKEA	VISHAY	Resistance, 0603, 13.3K/1%	R85, R90, R95, R100	4
CRCW06034K70FKEA	VISHAY	Resistance, 0603, 4.7K/1%	R87, R92, R97, R102	4
CRCW06031K20FKEA	VISHAY	Resistance, 0603, 1.2K/1%	R88, R93, R98, R103	4
CRCW06032K00FKEA	VISHAY	Resistance, 0603, 2K/1%	R89, R94, R99, R104	4
CRCW060360R4FKEA	VISHAY	Resistance, 0603, 60.4R/1%	R108, R109	2
RK09K1130A5R	ALPSALPINE	Compact type potentiometer, 9.8 mm Width	R110	1
1FS3T6B6M2QES−5	Dailywell	Miniature Toggle Switch	S1	1
SBC817−40LT1G	**onsemi**	NPN Transistor, SOT−23	T1, T2	2
NCV890200PDR2G	**onsemi**	Buck Switching Regulator, 2 A, SOIC−8	U1	1
LV8968BBUW	**onsemi**	Gate Driver, LQFP48−7 x 7	U2	1
NXV04V120DB1	**onsemi**	Automotive Power Module, APM19	U3	1
ACS758KCB−150B−PFF−T	ALLEGRO	Current Sensor	U4, U5	2
NCV20072DR2G	**onsemi**	Amplifier, 2−Channel, SOIC−8	U6, U10, U11	3
AD2S1210	ADI	Resolver Decoder, LQFP48−7 x 7	U7	1
CAT809LTBI	**onsemi**	ResetChip, SOT−23	U8	1
SC431BVSNT1G	**onsemi**	Voltage Reference, SOT−23	U9	1
NCV20074DR2G	**onsemi**	Amplifier, 4−Channel, SOIC−14	U12, U13	2
NCV7340D14R2G	**onsemi**	CAN Transceiver, SOIC−8	U14	1
XSMEELNANF−8.192 MHZ	TAITIEN	Crystal, 8.192MHz, 5032	Y1	1

PUBLICATION ORDERING INFORMATION

LITERATURE FULFILLMENT:
Email Requests to: orderlit@onsemi.com

onsemi Website: www.onsemi.com

TECHNICAL SUPPORT
North American Technical Support:
Voice Mail: 1 800−282−9855 Toll Free USA/Canada
Phone: 011 421 33 790 2910

Europe, Middle East and Africa Technical Support:
Phone: 00421 33 790 2910
For additional information, please contact your local Sales Representative