Version	Comment	Released
1	First release.	2018-08-12
2	Added support for Renesas HIP9011.	2018-10-06
3	Updated schematics and BOM.	2020-04-12

Table of contents

1 - Disclaimer

This device is designed for educational use. It is intended to be used by students, hobbyists and professionals. This device and the accompanying software are provided as is and should never be used in any circumstance where it is left unattended or could jeopardize the safety of an individual or property.

The purpose of this device is to give students, hobbyists and professionals the possibility to learn how to implement a programmable knock detection sensor in their products or projects. Enabling knock detection will assist in developing more efficient and environmentally friendly combustion-based solutions. All the necessary software examples to get started is available on our GitHub page.

2 – Compatibility

2.1 – Boards

This device or shield is designed and verified to be used with the Arduino Uno. If used with any other single-board microcontroller or computer, first ensure that it is pin and signal compatible.

2.2 - Sensors

All sensors of the type Bosch KS4 is compatible with this device. If you want to purchase a compatible sensor we recommend the KS4-P Bosch model 0 261 231 173.

3 – Operation

3.1 – Connection Table

Pin Sensor	Pin Knock Shield	Function
1	1	Signal Channel 1
2	2	GND
2	3	GND
1	4	Signal Channel 2

3.2 – Programming SPU

The signal processing unit (SPU) of this device require some initial information to give an accurate reading of the knock signal. The main function of the SPU is to not only read the levels of the sensor but also to analyze the frequency of which they occur. Matching the frequency to the combustion chamber is necessary, however this can be approximated as a starting value. Further analysis is however required for an optimal output.

3.2.1 – Setting Prescalar

This device features a dedicated 6 MHz oscillator. The SPU needs to be programmed with this information. By using the SPI interface, set the SPU prescalar by transferring the corresponding value of 0x44.

[SPI_TX - 0x44] (0b**01**000100)

The first bit (0b0100010**<u>0</u>**) sets if you want the SPI Slave output pin enabled. It will echo the input pin data if set to 0 or set to high impedance with 1.

3.2.2 – Setting Channel

By using the SPI interface, set the SPU active channel by transferring the corresponding value.

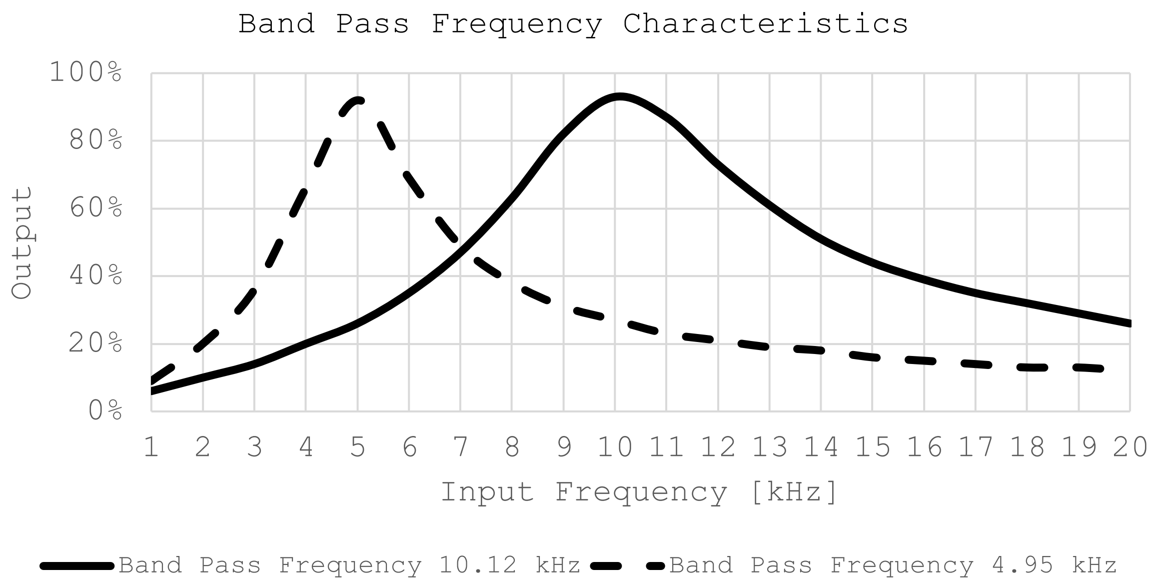
Channel 1: [SPI_TX – 0xE0] (0b**11**100000)
Channel 2: [SPI_TX – 0xE1] (0b**11**100001)

3.2.3 – Setting Band Pass Frequency

By using the SPI interface, set the band pass frequency by transferring the corresponding value. As an example, a band pass frequency of 7.27 kHz is set by transferring 0x2A to the SPU.

[SPI_TX – 0x2A] (0b**00**101010)

For more information see Appendix A and the SPU datasheet.



Input signal:	Sine wave 300 mVpp
Band pass frequency:	10.12 kHz / 4.95 kHz
Programmable gain:	0.200
Time constant:	100 µs
Measurement time:	3 ms

Note, an approximated knock frequency can be calculated. However, it should only be used as an initial value. Further evaluation and analysis are required for an accurate reading.

The following formula can be used to approximate the knock frequency of an engine based on its bore size.

$C = [m/s]$ Speed of sound can be approximated to 1200 m/s.
$\varnothing_{bore} = [mm]$ Bore size of cylinder in mm.

Knock frequency approximation in kHz:

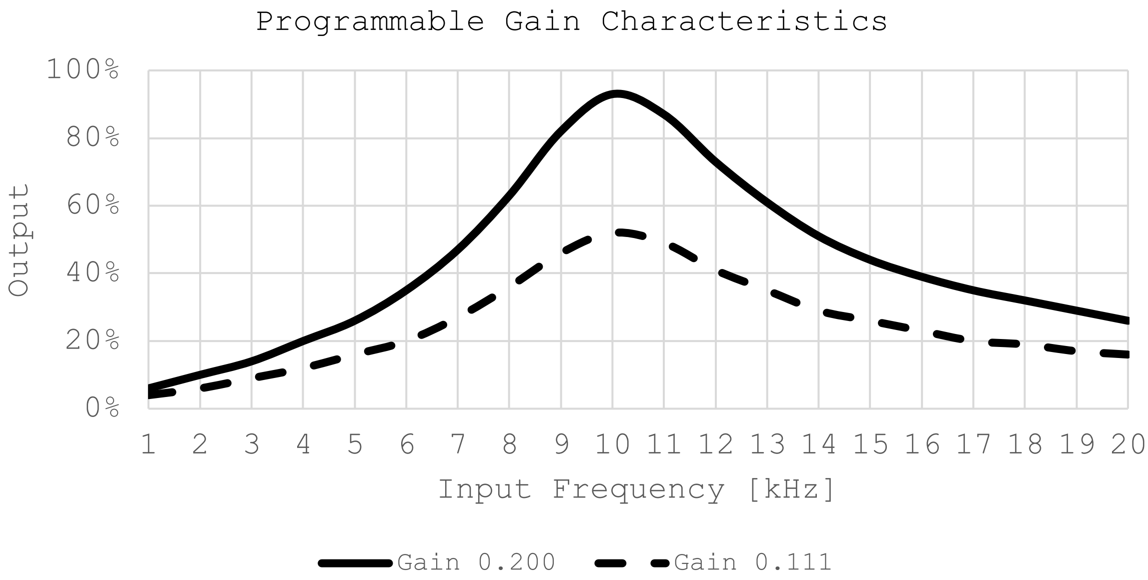
$$f_{calc} = \frac{3 \cdot C}{2 \cdot \pi \cdot \varnothing_{bore}}$$

3.2.4 – Setting Programmable Gain

The SPU features a signal gain amplifier, programmable by the SPI interface. The gain amplifier ranges between 0.111 – 2.000. Set the gain by transferring the corresponding value. As an example, a gain of 1.00 is set by transferring 0x8E to the SPU.

[SPI_TX – 0x8E] (0b**10**001110)

For more information see Appendix B and the SPU datasheet.



Input signal:	Sine wave 300 mVpp
Band pass frequency:	10.12 kHz
Programmable gain:	0.200 / 0.111
Time constant:	100 µs
Measurement time:	3 ms

Note, in addition to the programmable gain, the input amplifier gain can be adjusted by changing the value of R1, R2, R3 and R4. For convenience the Knock Shield is fitted with thru hole resistors for easy modification. Standard gain is 1x. See section 5 for more information.

The input amplifier gain is calculated for channel 1:

$$A_{IN} = \frac{R2}{R1}$$

For channel 2:

$$A_{IN} = \frac{R4}{R3}$$

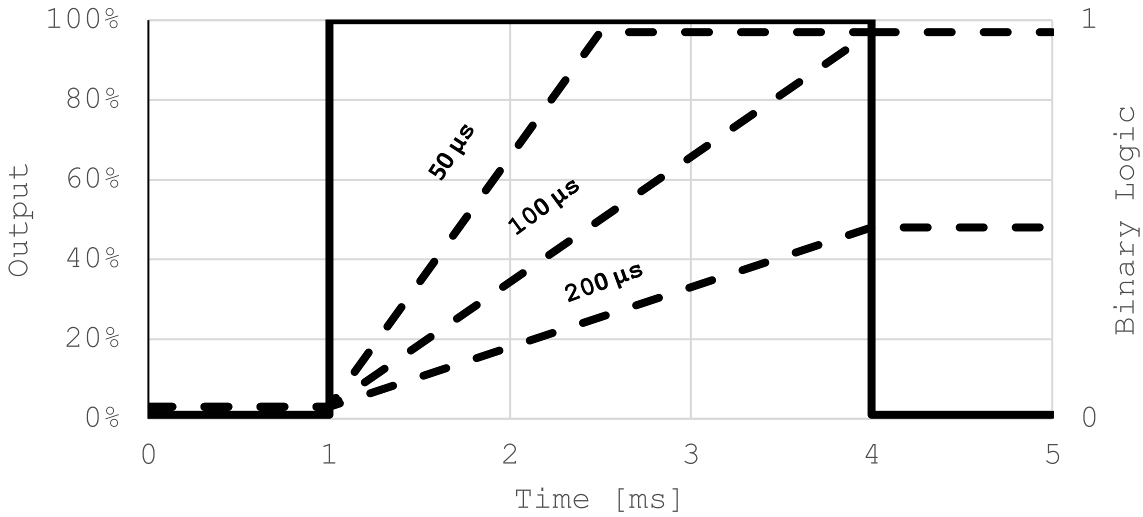
3.2.5 – Setting Integration Time Constant

The SPU features an integration time constant, programmable by the SPI interface. The faster you need to measure, the shorter the time constant need to be. This is suitable for when detecting knock on specific cylinders and the measurement time need to be in correlation with the engine speed. A time constant of 100 µs is set by transfering 0xCA to the SPU.

[SPI_TX – 0xCA] (0b**110**01010)

For more information see Appendix C and the SPU datasheet.

Time Constant Characteristics

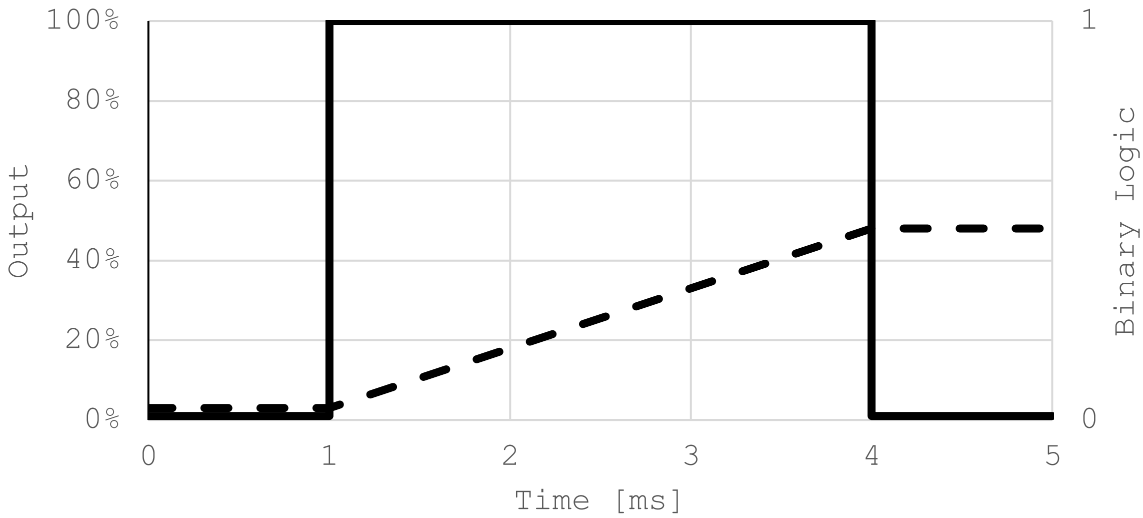


Input signal:	Sine wave 300 mVpp @ 10 kHz
Band pass frequency:	10.12 kHz
Programmable gain:	0.200
Time constant:	50 µs / 100 µs / 200 µs
Measurement time:	3 ms

3.3 – Measuring Knock Level

Before the measurements can start, the SPU needs to be programmed and the input channel need to be selected. See section 3.2.

Once the SPU is ready, it requires input on when to start and stop the measurement window.



The above example illustrates a typical operation of the SPU and it comprises of a series of necessary sequential operations.

1. The measurement window starts by driving digital pin 4 high.

2. The SPU performs the integration process and increases the output voltage based on the signal processing result.

3. The measurement window ends by driving digital pin 4 low.

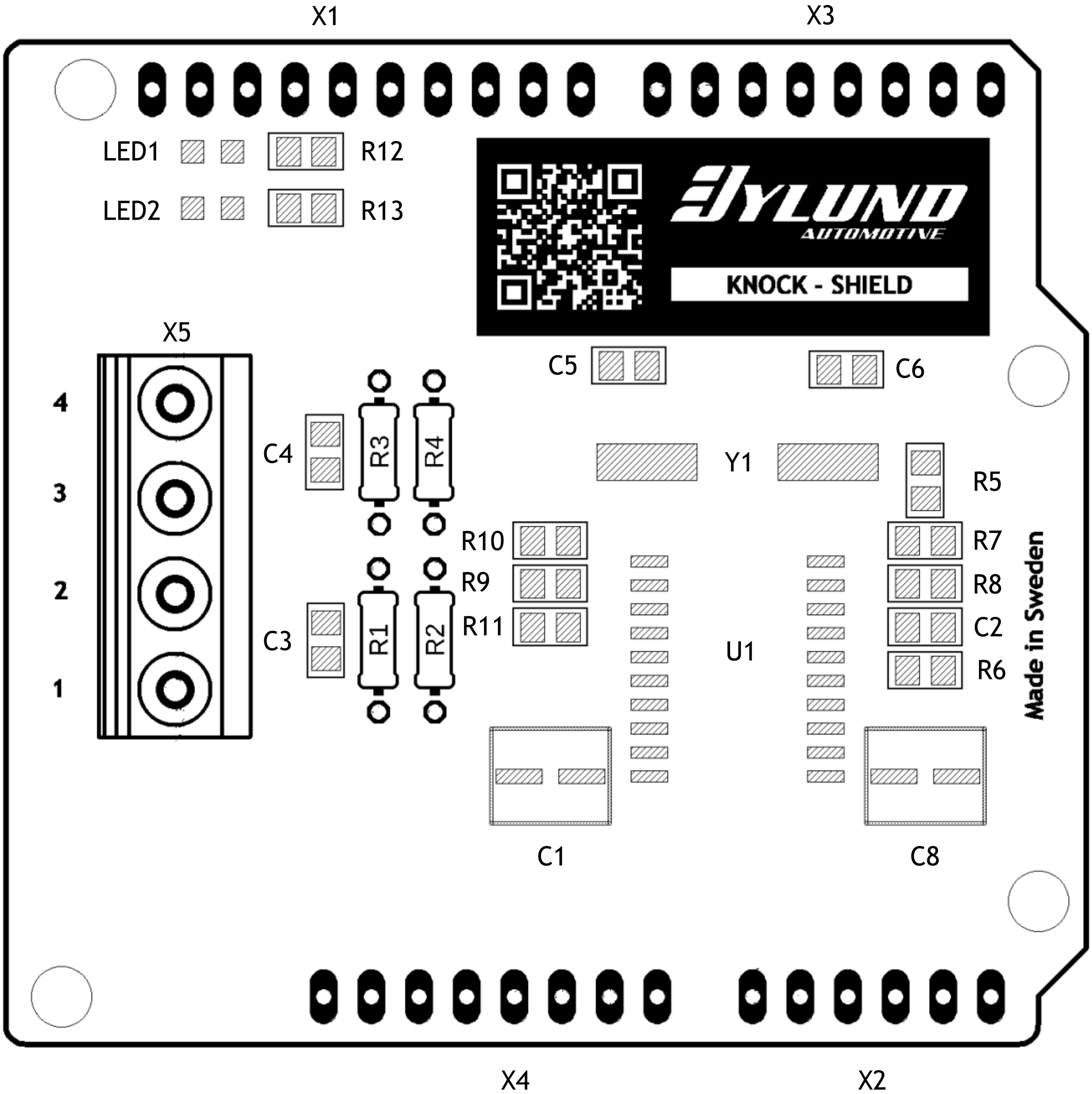
4. The SPU stops the integration process and the output voltage is frozen until the window starts again.

5. The SPU output voltage is read by the Arduino ADC on analogue input pin 0.

The measurement window is defined by your operation requirements and can be used in combination with an angular positioning sensor, sequential measuring and channel selection.

5 - Components

5.1 - Component Positions

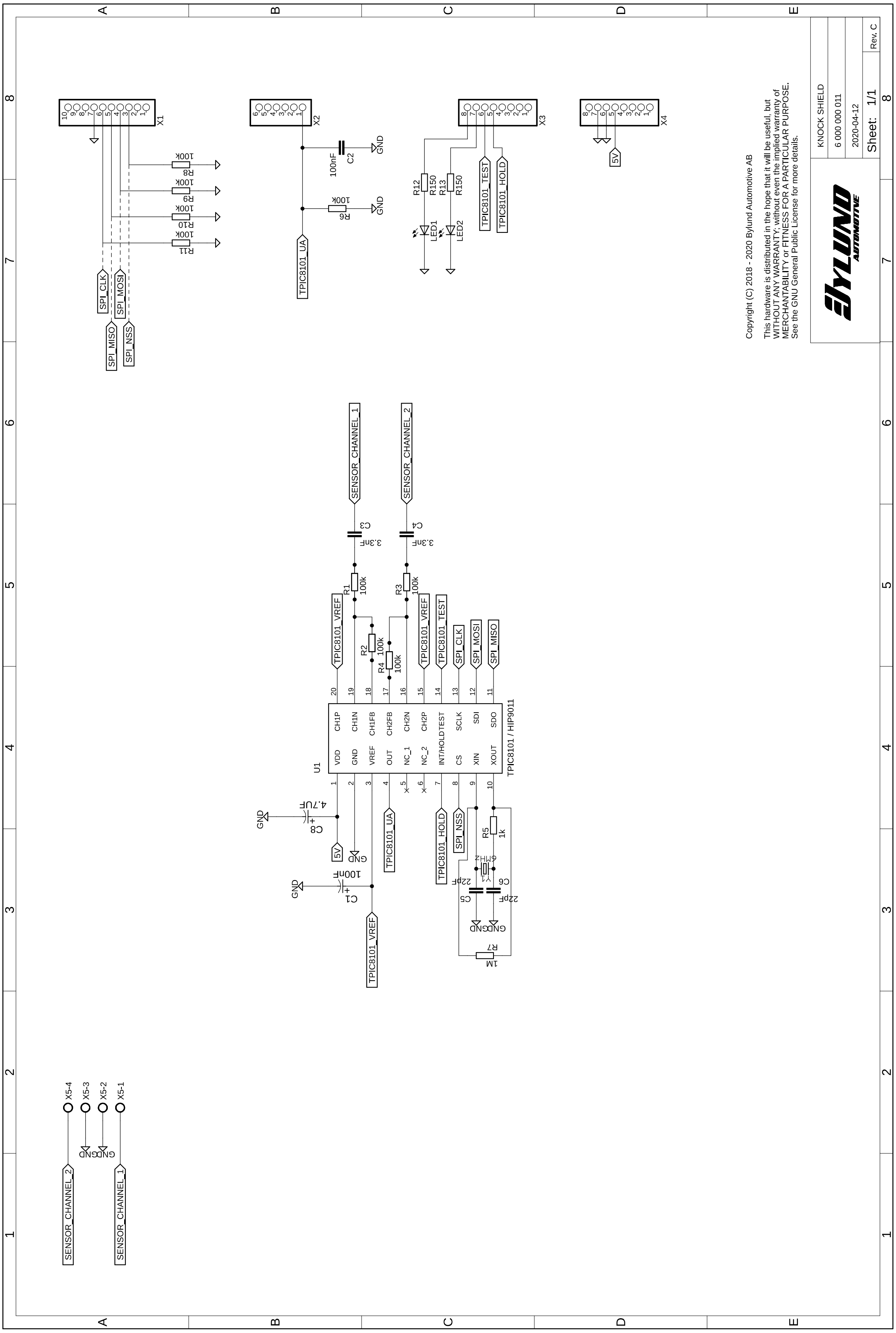


5.2 - Component List (BOM)

PCBA Component	Description
C1	100nF Electrolytic Capacitor
C2	100nF Capacitor C0805
C3, C4	3.3nF Capacitor C0805
C5, C6	22pF Capacitor C0805
C8	4.7uF Electrolytic Capacitor
R6, R8, R9, R10, R11	100k Resistor R0805
R1, R2, R3, R4	100k Resistor LR0204F / R0805
R5	1k Resistor R0805
R7	1M Resistor R0805
R12, R13	150R Resistor R0805
U1[1]	Knock Interface SPU
Y1	6MHz Oscillator
X1	Stackable header 10-pin
X2	Stackable header 6-pin
X3, X4	Stackable header 8-pin
X5	Terminal 4-pin
LED1, LED2	Green LED 0805

1. Texas Instruments part no. TPIC8101DW.
1. Renesas part no. HIP9011ABZ.

5.3 – Schematics



Appendix A - Programmable Band Pass Frequency Settings

Band Pass Frequency	Hexadecimal Value	Binary Value
1.22 kHz	0x00	0b**00**000000
1.26 kHz	0x01	0b**00**000001
1.31 kHz	0x02	0b**00**000010
1.35 kHz	0x03	0b**00**000011
1.40 kHz	0x04	0b**00**000100
1.45 kHz	0x05	0b**00**000101
1.51 kHz	0x06	0b**00**000110
1.57 kHz	0x07	0b**00**000111
1.63 kHz	0x08	0b**00**001000
1.71 kHz	0x09	0b**00**001001
1.78 kHz	0x0A	0b**00**001010
1.87 kHz	0x0B	0b**00**001011
1.96 kHz	0x0C	0b**00**001100
2.07 kHz	0x0D	0b**00**001101
2.18 kHz	0x0E	0b**00**001110
2.31 kHz	0x0F	0b**00**001111
2.46 kHz	0x10	0b**00**010000
2.54 kHz	0x11	0b**00**010001
2.62 kHz	0x12	0b**00**010010
2.71 kHz	0x13	0b**00**010011
2.81 kHz	0x14	0b**00**010100
2.92 kHz	0x15	0b**00**010101
3.03 kHz	0x16	0b**00**010110
3.15 kHz	0x17	0b**00**010111
3.28 kHz	0x18	0b**00**011000
3.43 kHz	0x19	0b**00**011001
3.59 kHz	0x1A	0b**00**011010
3.76 kHz	0x1B	0b**00**011011
3.95 kHz	0x1C	0b**00**011100
4.16 kHz	0x1D	0b**00**011101
4.39 kHz	0x1E	0b**00**011110
4.66 kHz	0x1F	0b**00**011111
4.95 kHz	0x20	0b**00**100000
5.12 kHz	0x21	0b**00**100001
5.29 kHz	0x22	0b**00**100010
5.48 kHz	0x23	0b**00**100011
5.68 kHz	0x24	0b**00**100100
5.90 kHz	0x25	0b**00**100101
6.12 kHz	0x26	0b**00**100110
6.37 kHz	0x27	0b**00**100111
6.64 kHz	0x28	0b**00**101000
6.94 kHz	0x29	0b**00**101001
7.27 kHz	0x2A	0b**00**101010
7.63 kHz	0x2B	0b**00**101011
8.02 kHz	0x2C	0b**00**101100
8.46 kHz	0x2D	0b**00**101101
8.95 kHz	0x2E	0b**00**101110
9.50 kHz	0x2F	0b**00**101111
10.12 kHz	0x30	0b**00**110000
10.46 kHz	0x31	0b**00**110001
10.83 kHz	0x32	0b**00**110010
11.22 kHz	0x33	0b**00**110011
11.65 kHz	0x34	0b**00**110100

12.10 kHz	0x35	0b**00**110101
12.60 kHz	0x36	0b**00**110110
13.14 kHz	0x37	0b**00**110111
13.72 kHz	0x38	0b**00**111000
14.36 kHz	0x39	0b**00**111001
15.07 kHz	0x3A	0b**00**111010
15.84 kHz	0x3B	0b**00**111011
16.71 kHz	0x3C	0b**00**111100
17.67 kHz	0x3D	0b**00**111101
18.76 kHz	0x3E	0b**00**111110
19.98 kHz	0x3F	0b**00**111111

Appendix B - Programmable Gain Settings

Programmable Gain	Hexadecimal Value	Binary Value
2.000	0x80	0b**10**000000
1.882	0x81	0b**10**000001
1.778	0x82	0b**10**000010
1.684	0x83	0b**10**000011
1.600	0x84	0b**10**000100
1.523	0x85	0b**10**000101
1.455	0x86	0b**10**000110
1.391	0x87	0b**10**000111
1.333	0x88	0b**10**001000
1.280	0x89	0b**10**001001
1.231	0x8A	0b**10**001010
1.185	0x8B	0b**10**001011
1.143	0x8C	0b**10**001100
1.063	0x8D	0b**10**001101
1.000	0x8E	0b**10**001110
0.944	0x8F	0b**10**001111
0.895	0x90	0b**10**010000
0.850	0x91	0b**10**010001
0.810	0x92	0b**10**010010
0.773	0x93	0b**10**010011
0.739	0x94	0b**10**010100
0.708	0x95	0b**10**010101
0.680	0x96	0b**10**010110
0.654	0x97	0b**10**010111
0.630	0x98	0b**10**011000
0.607	0x99	0b**10**011001
0.586	0x9A	0b**10**011010
0.567	0x9B	0b**10**011011
0.548	0x9C	0b**10**011100
0.500	0x9D	0b**10**011101
0.471	0x9E	0b**10**011110
0.444	0x9F	0b**10**011111
0.421	0xA0	0b**10**100000
0.400	0xA1	0b**10**100001
0.381	0xA2	0b**10**100010
0.364	0xA3	0b**10**100011
0.348	0xA4	0b**10**100100
0.333	0xA5	0b**10**100101
0.320	0xA6	0b**10**100110
0.308	0xA7	0b**10**100111
0.296	0xA8	0b**10**101000
0.286	0xA9	0b**10**101001
0.276	0xAA	0b**10**101010
0.267	0xAB	0b**10**101011
0.258	0xAC	0b**10**101100
0.250	0xAD	0b**10**101101
0.236	0xAE	0b**10**101110
0.222	0xAF	0b**10**101111
0.211	0xB0	0b**10**110000
0.200	0xB1	0b**10**110001
0.190	0xB2	0b**10**110010
0.182	0xB3	0b**10**110011
0.174	0xB4	0b**10**110100
0.167	0xB5	0b**10**110101
0.160	0xB6	0b**10**110110

0.154	0xB7	0b**10**110111
0.148	0xB8	0b**10**111000
0.143	0xB9	0b**10**111001
0.138	0xBA	0b**10**111010
0.133	0xBB	0b**10**111011
0.129	0xBC	0b**10**111100
0.125	0xBD	0b**10**111101
0.118	0xBE	0b**10**111110
0.111	0xBF	0b**10**111111

Appendix C - Programmable Integrator Time Constants

Time	Hexadecimal Value	Binary Value
40 µs	0xC0	0b**110**00000
45 µs	0xC1	0b**110**00001
50 µs	0xC2	0b**110**00010
55 µs	0xC3	0b**110**00011
60 µs	0xC4	0b**110**00100
65 µs	0xC5	0b**110**00101
70 µs	0xC6	0b**110**00110
75 µs	0xC7	0b**110**00111
80 µs	0xC8	0b**110**01000
90 µs	0xC9	0b**110**01001
100 µs	0xCA	0b**110**01010
110 µs	0xCB	0b**110**01011
120 µs	0xCC	0b**110**01100
130 µs	0xCD	0b**110**01101
140 µs	0xCE	0b**110**01110
150 µs	0xCF	0b**110**01111
160 µs	0xD0	0b**110**10000
180 µs	0xD1	0b**110**10001
200 µs	0xD2	0b**110**10010
220 µs	0xD3	0b**110**10011
240 µs	0xD4	0b**110**10100
260 µs	0xD5	0b**110**10101
280 µs	0xD6	0b**110**10110
300 µs	0xD7	0b**110**10111
320 µs	0xD8	0b**110**11000
360 µs	0xD9	0b**110**11001
400 µs	0xDA	0b**110**11010
440 µs	0xDB	0b**110**11011
480 µs	0xDC	0b**110**11100
520 µs	0xDD	0b**110**11101
560 µs	0xDE	0b**110**11110
600 µs	0xDF	0b**110**11111