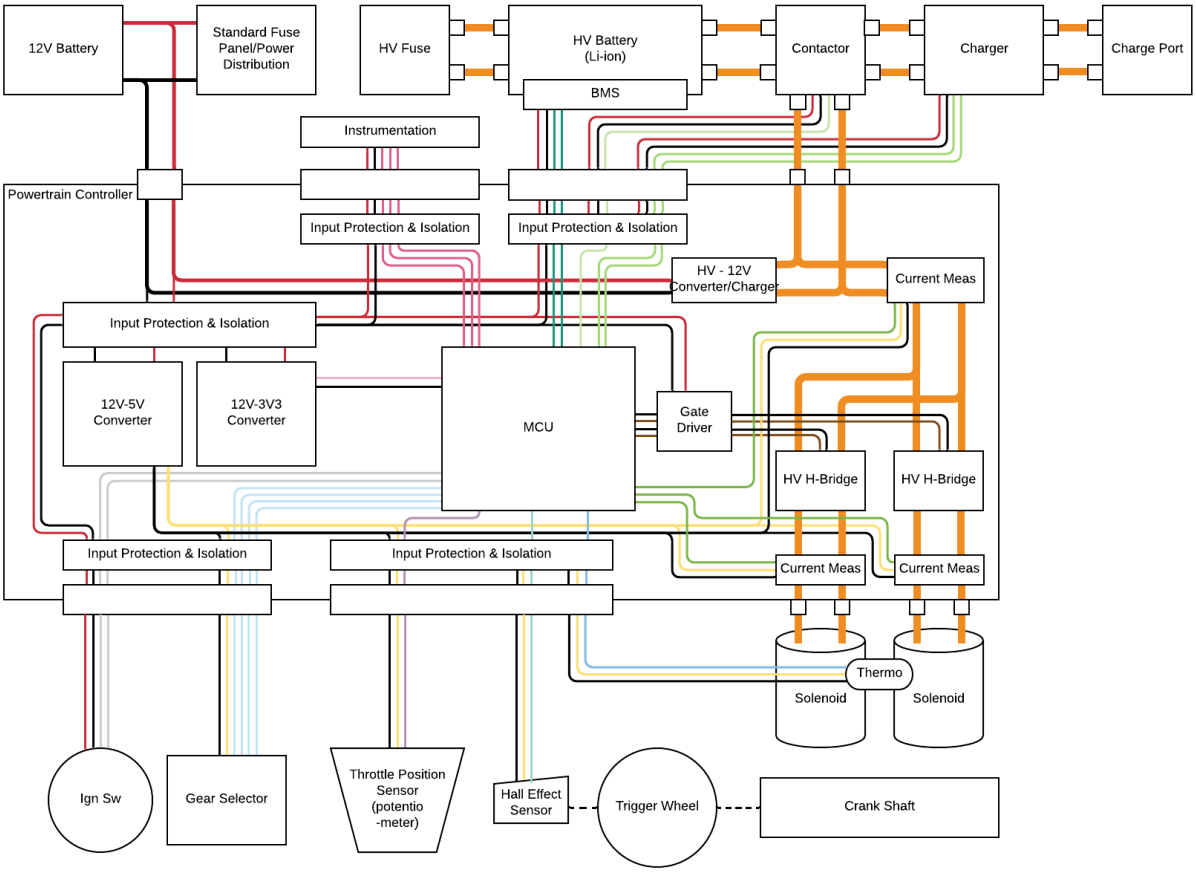
Electric Engine Conversion



20 July 2019

Proprietary and Confidential

Prepared by

Nick Anderson – Electrical Engineer

Shane Vogt – Mechanical Engineer

MMCKENNA – ELECTRIC ENGINE

Feasibility Study

PRODUCT DESCRIPTION

The final version of this product is meant to be a kit which allow for combustion engines to be converted into electric powertrains. The system will utilize linear electromagnets mounted at the top of each cylinder with a permanent magnet mounted on each piston. The electromagnets are timed and activated in such way to replicate the behavior of combustion. This timing will be computer controlled based on the crank angle of the engine.

Alternatively this system can be adapted to be a complete drop in engine.

SCOPE OF STUDY

Because the inventor has a very good handle on the mechanics of this system, and has proven some preliminary success, this feasibility study will be geared towards guiding the design and development of the control electronics of the system, especially the prototyping and testing phases. Additionally, we will comment on the expected performance of the system given the selected components. We will perform some rudimentary calculations, but proper simulation and evaluation of the system is complex and outside the scope of this study.

SUMMARY

In this study we evaluate and recommend prototyping friendly components which can be configured without special knowledge or equipment and explain how to use them. We also outline the firmware and make recommendations. The system we are recommending is a dual channel high-side driven single polarity electromagnet motorcontroller, which utilizes DC solid state relays for switching. For sensors we recommend using two optical photo interrupters sensing slots on a trigger wheel affixed to the crankshaft, along with a slide potentiometer to simulate a throttle position. The programmable controller for the system should be an Arduino UNO MCU dev board.

We recommend programming everything in unitless 8 bit variables, utilizing a look up table to control the angle after top dead center which the electromagnets are activated as well as the duration that they are activated for. The inputs for the look up table will be the measured RPM and the throttle position sensor. The electromagnets will be timed to the crankshaft position by counting pulses after the top dead center pulse. The RPM will be calculated by measuring the number of pulses that are sensed in a 50 millisecond period.

We recommend utilizing an excel spreadsheet to translate real-world units to the unit-less variables stored in the MCU. The 8 bit variables can process very quickly and reliably within the microcontroller and will provide enough resolution for prototyping.

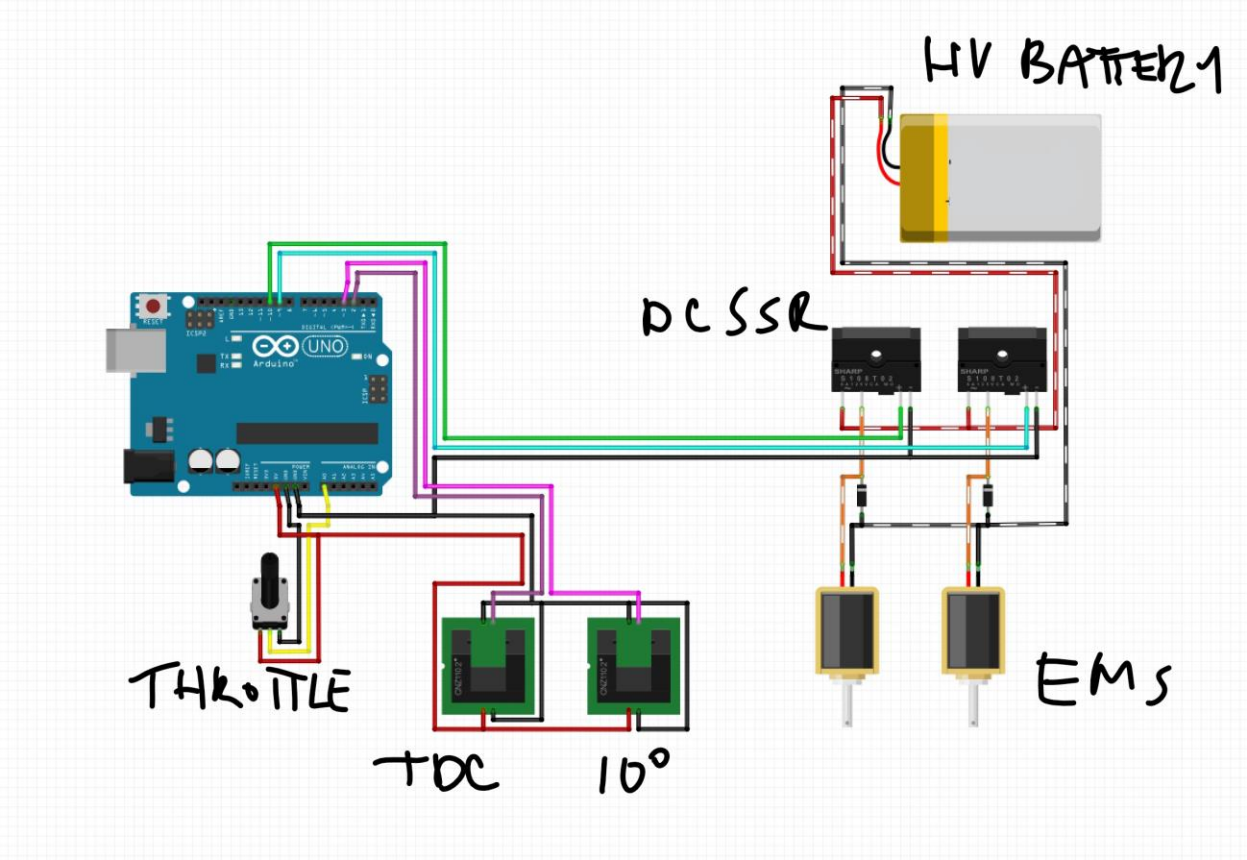
We do believe that for a production version of the invention, stronger electromagnets will need to be used that can handle many kilowatts of power. Additionally, the production version will likely benefit from dual coil electromagnets that can “push” AND “pull” the permanent magnets, as a way to double the duty cycle.

The image on the cover page outlines what the production version of the device might look like, and the images below outline what the minimum viable prototyping version should look like. We believe that the next prototyping step for the customer is what we’ve outlined below.

applied design 5311 Western Ave., Suite 131 - Boulder, CO - 80301 - p.303-444-3125 - f.303-444-3126 - www.appliedesigncorp.com

CONTROL ELECTRONICS

The control electronics are essentially an adaptation of a standard motorcontroller for controlling a brushless AC reluctance motor (permanent magnet motor), but instead of controlling current through the poles of the motor in a circular pattern, the poles are arranged in a line. For a production version of the invention, the easiest path to developing the control electronics will be to modify an existing high voltage motorcontroller, or to partner with a motorcontroller company. For testing and prototyping purposes we recommend the following setup:



A microcontroller in the form of an Arduino UNO (or similar) acts as the brains of the motorcontroller. It receives three signals; a throttle position in the form of a potentiometer and two signals from optical interrupters to determine crank position (more on these later). Based on those signals it drives the two electromagnets (EM) via high-voltage DC solid state relays (DC SSR) in sync with the crank position. The control input of the solid state relay is electrically isolated from the high-voltage drive circuit ensuring that the sensitive low-voltage control electronics are safe.

1. DC Solid State Relays

In a production setting, MOSFET transistors integrated into a custom motorcontroller will provide the most reliability, efficiency, and enable additional functionality, such as regenerative braking. However, there are not readily available MOSFET modules which can handle the high voltages of this device, so we recommend using solid state relays (SSRs), which are simple to use and can handle high voltages and loads. Solid state relays come in both DC and AC variants, many AC solid state relays employ a switch-on-zero scheme where the SSR will only switch on-to-off or off-to-on when the AC voltage crosses zero. When used with DC voltages, it may never switch on/off, therefore it is important to source DC solid state relays.

applied design 5311 Western Ave., Suite 131 - Boulder, CO - 80301 - p.303-444-3125 - f.303-444-3126 - www.appliedesigncorp.com

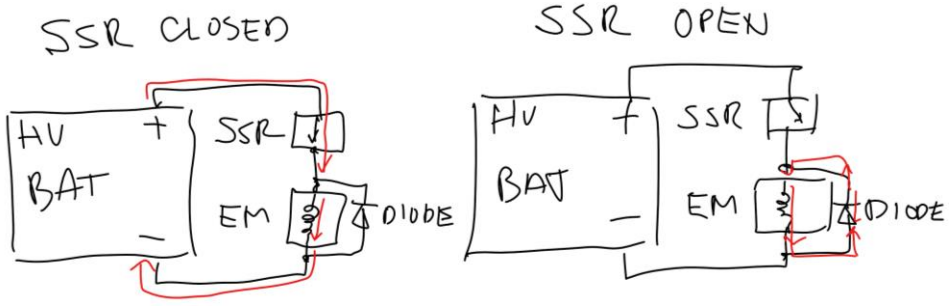
A good option for prototyping is the TE SSRDC-200D25 which can be purchased from DigiKey for $48.30. It is rated to deliver 200 VDC at 25 A, which should give enough margin for the 120 VDC and 12 A that the electromagnets (Ems) are drawing. If the electromagnet's polarity is not being switched, then one DCSSR will be required per electromagnet, but if the polarity must be controllable, then 4 DCSSRs will be required per electromagnet arranged in an h-bridge configuration.

It's recommended that these DCSSRs be mounted to a heatsink, such as the Sensata-Crydom HS251, which can be purchased from DigiKey for $17.18, which also requires a thermal pad such as HSP-2, available from DigiKey for $1.33.

The DCSSR can control the EM either as a high-side or low-side switch, meaning the DCSSR can either be placed between the EM and the HV-battery on the positive or negative side. It's just important to wire the SSR in a way that current flows from the positive terminal to the negative terminal, in the case of the above SSR, from terminal 2 to 1.

The SSR is controlled when a DC voltage of 3.5 V to 32 V is applied across the input terminals, 3 and 4 in the above example. The selected microcontroller can drive these inputs directly from their output pins, which supply 5 V. The SSR conducts electricity when 5 V is applied and does not conduct when 0 V is applied. The negative side of this input should be connected to the Arduino's ground, and the positive side should be connected to a pin supporting PWM, in the drawings pins 9 and 10 are selected.

The SSR as well as most power electronics don't handle inductive loads like an electromagnet very well. An inductor (which an electromagnet is a large version of) store energy in the form of an electromagnetic field, which will release energy back into the circuit when the power is removed. This manifests as a large voltage spike on the negative side of the electromagnet when power is removed from it, which is why you will see sparks when working with EMs. To prevent this voltage spike from damaging the DCSSR a snubber diode must be used to allow the energy to be dissipated. To do so a large Schottky diode must be placed across the input of the EM, such that it will conduct from the negative to positive sides. When the SSR is open and switches off the power to the EM, a positive voltage will form on the negative side of the EM, and negative voltage on the positive side. This will cause the diode to conduct current from the bottom of the EM to the top and keep the SSR safe.



applied design 5311 Western Ave., Suite 131 - Boulder, CO - 80301 - p.303-444-3125 - f.303-444-3126 - www.appliedesigncorp.com

The diode must be fairly large as it will dissipate that energy in the form of heat. The DST2045AX is a good option and can be purchased from DigiKey for $1.35.

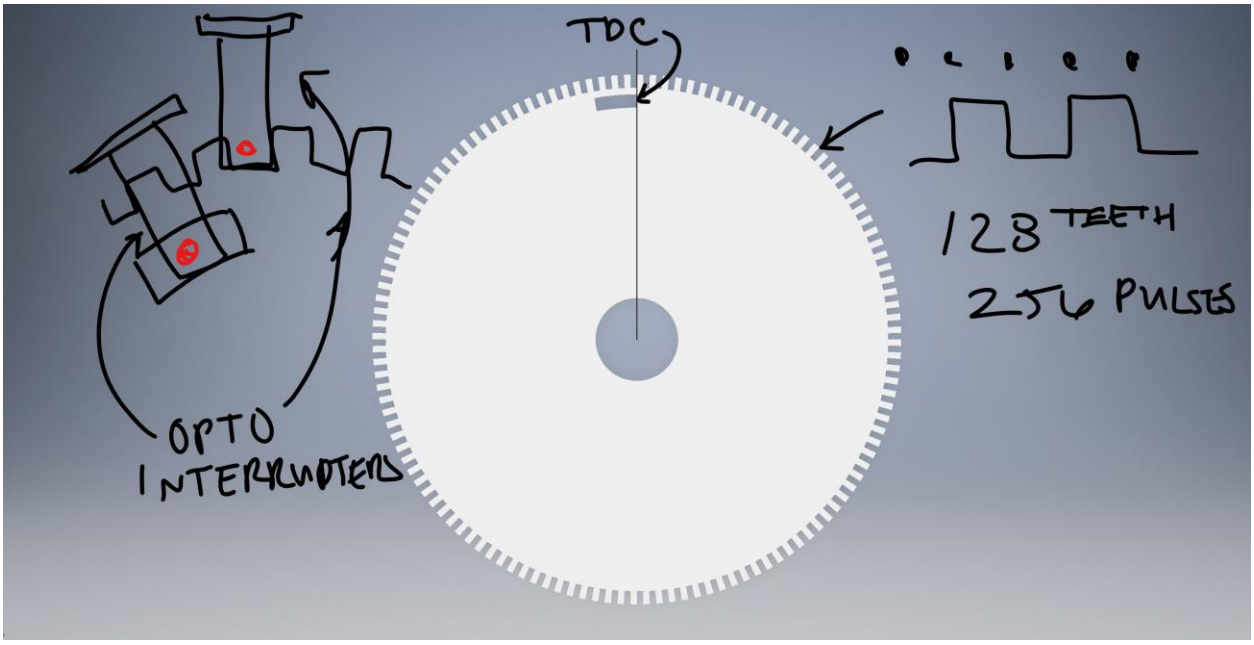
2. Crank Angle Sensor

In production, hall-effect sensors and trigger wheels are considered the most reliable way to determine the angular position of a crankshaft. But hall effect sensors require a bit of additional signal processing, so we propose using two optical interrupters in conjunction with a slotted trigger wheel.

We recommend using the GP1A57HRJ00F photo interrupters from SparkFun, along with the breakout board, and required 330 ohm resistor. The photo interrupter works by projecting a beam of light across a gap. A phototransistor on the other side of the gap will emit a voltage when the light is not blocked and go to zero when blocked.

By attaching a metal disk to the crank shaft and slotting it, the optical interrupted will generate a square wave as the crank rotates. The microcontroller can be configured to sense each transition of the square wave. Each one of these transitions will indicate a number of degrees traveled depending on the number of teeth on the trigger wheel. In the example below, there are 128 teeth with 256 transitions. This number was selected because the microcontroller is 8 bit architecture, meaning naturally processes 8 bit numbers, which has a value of 256. At 256 values, each transition is 1.4 degrees, which should provide enough resolution for this application. It is possible to get finer resolution if necessary, by counting time between the pulses relative to the calculated crank speed.

These 128 teeth will provide the microcontroller with the relative crank angle, meaning by counting pulses it will know how far the crankshaft has moved, but is unaware of where top dead center (TDC) is. To get the absolute position, an additional optical interrupter is needed as well as another slot in one location to indicate where TDC is (or any other angle relative to TDC). The microcontroller can be configured to only sense the falling edge of this signal to indicate TDC. This will reset the tooth count of the primary interrupter, making all measurements relative to TDC. This slot will also be used to calculate the RPM of the engine, by measuring the time between TDC pulses.



applied design 5311 Western Ave., Suite 131 - Boulder, CO - 80301 - p.303-444-3125 - f.303-444-3126 - www.appliedesigncorp.com

Physically, the relative teeth will be at the outer radius of the trigger wheel, and the TDC slot will be further in. The interrupters have a 12 mm depth, but only sense at the last 4 mm. The relative teeth will therefore need to be 4 mm deep, and the TDC slot will need to start at 6 mm deep and end at 10 mm deep, leaving 2 mm of clearance.

To interface with these interrupters they must be soldered to the breakout board with the resistor, and 5V and ground must be supplied to the board. The output of these interrupters should be connected to an Arduino pin that supports interrupts, on the UNO these are pins 2 and 3. These pins can be configured to run code on a rising, falling, or both rising and falling edges, and will be the bases of timing the control of the SSRs.

3. Throttle

For testing purposes we recommend using a potentiometer or slide pot to input a simulated throttle position. We recommend the COM-09119 from SparkFun because it is large and will provide a good visual of what position the throttle is in.

A potentiometer will put out a voltage proportional to the percent position of the slide. In this case 0% will be 0 V and 100% will be 5 V. To use, hook up pin 1 to the Arduinos 5V, pin 3 to the Arduinos GND, and pin 2 to a pin which can read an analog voltage, in the diagram above that's pin A0.

This combined with the calculated RPM will be the primary inputs into the timing function of the SSRs.

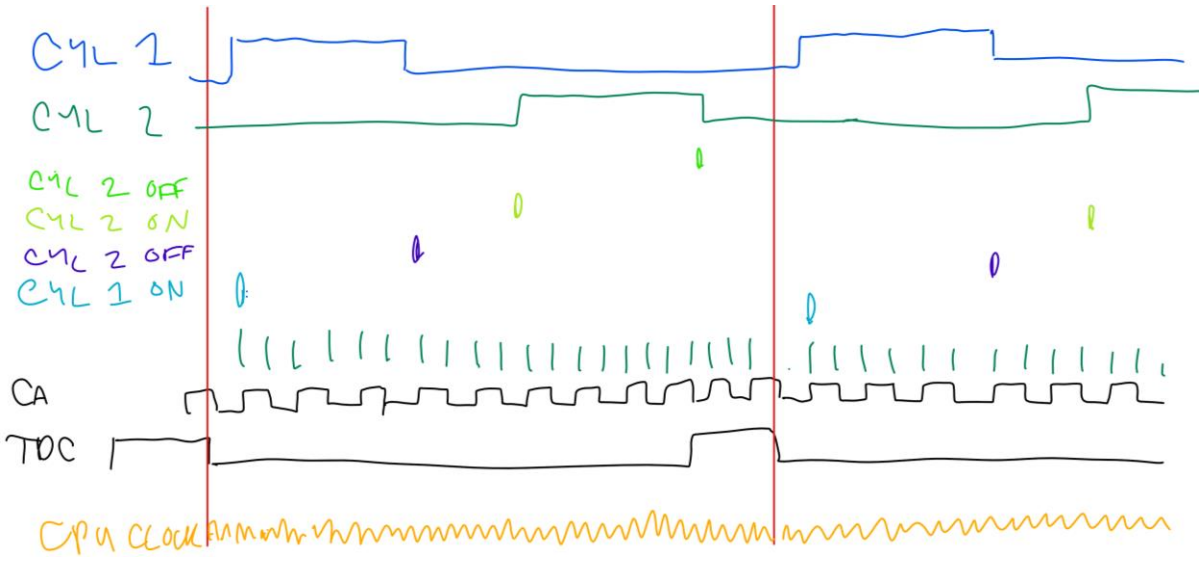
4. Microcontroller

A microcontroller (MCU) is a small programmable IC, which is much like a small computer. The easiest way to use a microcontroller in a prototype is by using a dev board, which houses the MCU as well as other supporting electronics. A favorite is the Arduino UNO, which can be purchased from SparkFun. It will need a USB cable for programming, and a power supply to run when not connected to the computer.

It uses the free and open source Arduino IDE which can be downloaded and installed here.

applied design 5311 Western Ave., Suite 131 - Boulder, CO - 80301 - p.303-444-3125 - f.303-444-3126 - www.appliedesigncorp.com

FIRMWARE



The purpose of the firmware will be to turn the EMs on and off in a way that's synchronized to the position of the crankshaft. The start and stop position of the EMs being turned on will be variables that are determine by the position of the throttle and the RPM of the engine. Internally, everything will be thought of and controlled in terms of number of crank-pulses, which will be 256 per revolution. Time will be measured by counting the number of CPU clock cycles, which are 16,000,000 per second for the Arduino UNO's ATMega328p MCU. Conversions to other measuring systems, such as degrees and seconds, will be done when values are input by the user or displayed to the user.

There will be 4 interrupt routines, one for the crank angle pulses, and one for the TDC pulses. The crank angle interrupts will occur on both the rising and falling edges of the square wave produced by the crank angle optical interrupter. The TDC interrupt will be configured to just occur on the falling edge of the square wave. Additionally, a special interrupt will occur every 0.050 second (20 Hz) based on CPU clock cycles to calculate RPM, and another that will occur every 0.010 second (100 Hz) to update the throttle position variable and look up table.

5. Global Variables

- crankAngle – an unsigned 8-bit variable storing the position of the crank, 0-255 = 0-360 deg
- crankSubPulses – an unsigned 8-bit variable storing the number of crank pulses to be divided by 8
- crankPulses – an unsigned 8-bit variable storing the number of crank pulses divided by in 50 ms, 0-255 = 0-255 Hz = 0-15,300 RPM
- crankVelocity – an unsigned 8-bit variable storing the RPM of the crank, 0-255 = 0-255 Hz = 0-15,300 RPM
- throttlePosition – an unsigned 8-bit variable storing the throttle position, 0-255 = 0-100%
- cyl1StartAngle – an unsigned 8-bit variable storing the position of the where the cyl 1 EM will be energized, 0-255 = 0-360 deg
- cyl1Duration – an unsigned 8-bit variable storing distance after the cyl1StartPosition for the cyl 1 EM will be deenergized, 0-255 = 0-360 deg
- cyl2StartAngle – an unsigned 8-bit variable storing the position of the where the cyl 2 EM will be energized, 0-255 = 0-360 deg
- cyl2Duration – an unsigned 8-bit variable storing distance after the cyl2StartPosition for the cyl 2 EM will be deenergized, 0-255 = 0-360 deg
- look up table with an array of start positions and durations, based on crankVelocity and throttlePosition

applied design 5311 Western Ave., Suite 131 - Boulder, CO - 80301 - p.303-444-3125 - f.303-444-3126 - www.appliedesigncorp.com

- cyl1Angle – an unsigned 8-bit variable storing the position of cyl 1 relative to the cyl 1 start position, this also acts as a flag indicating the EM should be on, if greater than 0
- cyl2Angle – an unsigned 8-bit variable storing the position of cyl 1 relative to the cyl 2 start position, this also acts as a flag indicating the EM should be on, if greater than 0

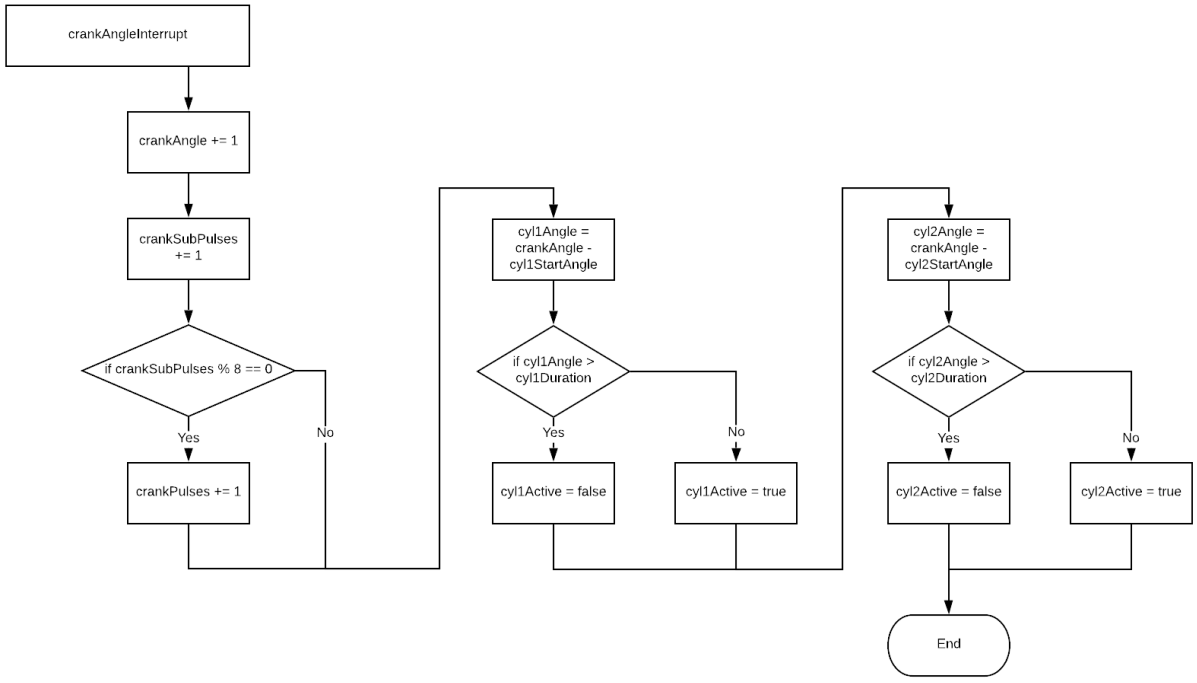
6. Crank Angle Interrupt

The crank angle interrupt is triggered on the falling and rising edge of the crank angle optical interrupter output. In this routine the crank angle is measured by counting the pulses, and it determines if the electromagnets should be activated or not. Additionally, pulses are counted and divided by 8 for use in calculating crankVelocity (RPM).

When the interrupt occurs, first the crankAngle variable is incremented to get its new position, followed by the crankSubPulses. The crankSubPulses are used to count every 8 pulses, before incrementing the crankPulses variable used in RPM calculation. This is done by taking the modulo of the crankSubPulses, and then checking for 0 (when 8 rolls over), if it is the crankPulses is incremented and is 1/8th the actual pulses.

Then the crankAngle position is converted to reference for of each cylinder based on what position it is supposed to active at by subtracting the cylXStartAngle from the crankAngle. The primary purpose of this is to allow firing sequence to cross TDC without needing extra cycles to wrap the angle. This logic assumes that the unsigned 8-bit variable that that cylXAngle is stored in will wrap, meaning, for example, if the current crankAngle is 10, and the cyl1StartAngle is 20, crankAngle minus cyl1StartAngle will equal 246 and not -10. If this is not the case, 256 will need to be added to the number, and the modulo of 256 will need to be taken to wrap this number correctly.

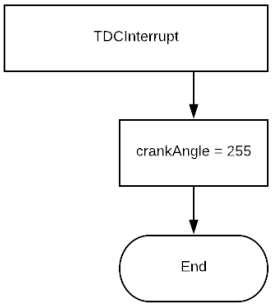
Once the cylinder angles are determined the EMs are activated if that angle is within the duration variable. The cylXStartAngle and cylXDuration are determined in a look up table dependent on crankVelocity (RPM) and throttlePosition.



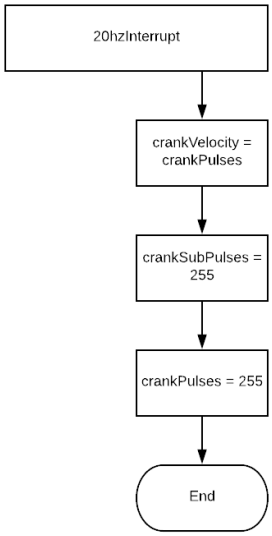
applied design 5311 Western Ave., Suite 131 - Boulder, CO - 80301 - p.303-444-3125 - f.303-444-3126 - www.applieddesigncorp.com

7. TDC Interrupt

The TDC interrupt is triggered on the falling edge of TDC optical interrupter, and marks the TDC location of the crank. This routine serves the purpose of resetting the crankAngle to 0, but does so by setting it to 255, which will become 0 on the next crankAngleInterrupt. The TDC interrupt should be arranged physically, so it gets triggered between the pulse 255 (last pulse) and tooth 0 (first pulse), with pulse 0 being true TDC.



8. 20 Hz Interrupt



This interrupt is configured by using a special interrupt routine that divides the number of clock cycles of the MCU by a prescaler, and interrupts when that value exceeds a threshold. To achieve this we will use Timer0, which supports a 16 bit threshold, divide the clock by the maximum prescaler of 1024, and trigger on 780 cycles. This will give us a routine frequency of 20 Hz.

In this routine we'll update the crankVelocity variable and reset the crankPulses and crankSubPulses variables which were used to count the number of pulses produced by the crank angle optical interrupter, divided by 8 to make fit in a 8 bit variable. The two variables get reset to 255 so they overflow to 0 on the next crankAngleInterrupt

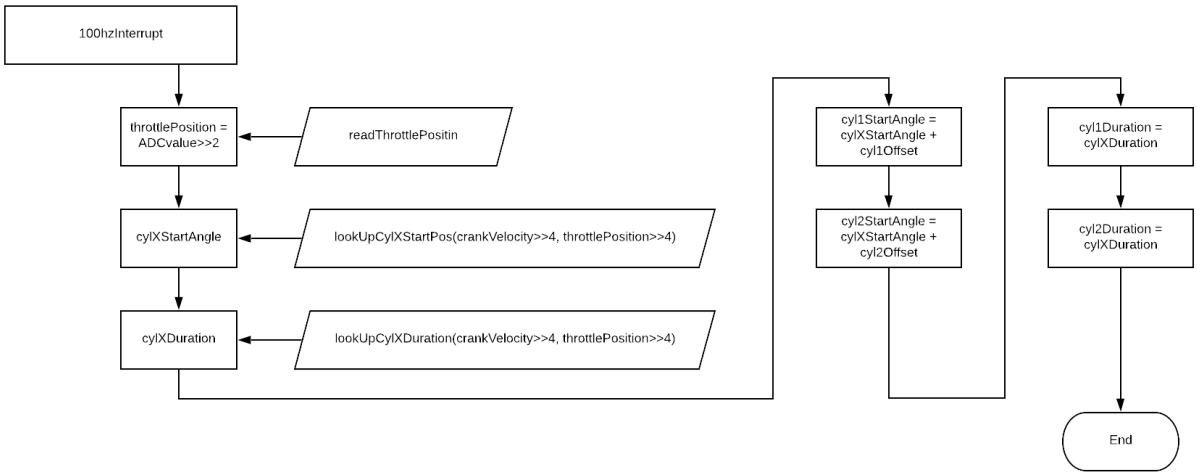
9. 100 Hz Interrupt

We read the throttle value and update the variables being fed into the cylXStart and cylXDuration variables based on the throttle position and crankVelcoity (RPM) in this interrupt routine. We call this routine more often than the 20 Hz interrupt routine because the calculating of RPM needs a longer sampling duration to be accurate. Additionally it is important for throttle response characteristics to be sampled quickly.

We read the throttle position from the ADC, which puts out a 10 bit value, so we must shift the output by 2 bits to get an 8 bit reading. We then reduce both the crankVelocity and throttlePosition to 4 bits to reduce our look up table size to 16 x 16, for 256 cell look up table, with two values each (cylXStartAngle, cylXDuration). Requiring 512 bytes of memory, which will fit in any memory bank in the MCU.

Once those values are looked up they are fed into the cyl1StartAngle and cyl2StartAngle by applying an offset (the first cylinder being located at 0 (0 degress) and the second being located at 127 (180 degrees)).

applied design 5311 Western Ave., Suite 131 - Boulder, CO - 80301 - p.303-444-3125 - f.303-444-3126 - www.appliedesigncorp.com



10. Look Up Tables

The look up tables will require manual 'tuning' running the engine at various speeds and testing start position and duration. Internally the tables are 16 columns wide and 16 rows deep, labeled 0-15, 0-15, with value 0-255. To translate real world units use the provided spreadsheet. Below are example variables in the look up tables, in both human readable, and machine readable form (machine readable being what is stored internally in the MCU).

Human Readable

		Throttle Position %															
ATDC (deg)		0%	7%	13%	20%	27%	33%	40%	47%	53%	60%	67%	73%	80%	87%	93%	100%
RPM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1200	0	6	5	4	4	3	3	2	2	2	1	1	1	1	1	1
	2400	0	7	6	5	5	4	4	3	3	3	2	2	2	2	2	2
	3600	0	8	7	6	6	5	5	4	4	4	3	3	3	3	3	3
	4800	0	9	8	7	7	6	6	5	5	5	4	4	4	4	4	4
	6000	0	10	9	8	8	7	7	6	6	6	5	5	5	5	5	5
	7200	0	11	10	9	9	8	8	7	7	7	6	6	6	6	6	6
	8400	0	12	11	10	10	9	9	8	8	8	7	7	7	7	7	7
	9600	0	13	12	11	11	10	10	9	9	9	8	8	8	8	8	8
	10800	0	14	13	12	12	11	11	10	10	10	9	9	9	9	9	9
	12000	0	15	14	13	13	12	12	11	11	11	10	10	10	10	10	10
	13200	0	16	15	14	14	13	13	12	12	12	11	11	11	11	11	11
	14400	0	17	16	15	15	14	14	13	13	13	12	12	12	12	12	12
	15600	0	18	17	16	16	15	15	14	14	14	13	13	13	13	13	13
	16800	0	19	18	17	17	16	16	15	15	15	14	14	14	14	14	14
	18000	0	20	19	18	18	17	17	16	16	16	15	15	15	15	15	15

Machine Readable

		throttlePosition>>4 (8-bit /16)															
cylXStartAngle (pulses ATDC)		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
crankVelocity >> 4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	8	7	5	5	4	4	2	2	2	1	1	1	1	1	1

applied design 5311 Western Ave., Suite 131 - Boulder, CO - 80301 - p.303-444-3125 - f.303-444-3126 - www.appliedesigncorp.com

2	0	9	8	7	7	5	5	4	4	4	2	2	2	2	2	2
3	0	11	9	8	8	7	7	5	5	5	4	4	4	4	4	4
4	0	12	11	9	9	8	8	7	7	7	5	5	5	5	5	5
5	0	14	12	11	11	9	9	8	8	8	7	7	7	7	7	7
6	0	15	14	12	12	11	11	9	9	9	8	8	8	8	8	8
7	0	16	15	14	14	12	12	11	11	11	9	9	9	9	9	9
8	0	18	16	15	15	14	14	12	12	12	11	11	11	11	11	11
9	0	19	18	16	16	15	15	14	14	14	12	12	12	12	12	12
10	0	21	19	18	18	16	16	15	15	15	14	14	14	14	14	14
11	0	22	21	19	19	18	18	16	16	16	15	15	15	15	15	15
12	0	23	22	21	21	19	19	18	18	18	16	16	16	16	16	16
13	0	25	23	22	22	21	21	19	19	19	18	18	18	18	18	18
14	0	26	25	23	23	22	22	21	21	21	19	19	19	19	19	19
15	0	28	26	25	25	23	23	22	22	22	21	21	21	21	21	21

Human Readable

Duration (deg)	Throttle Position %															
RPM	0%	7%	13%	20%	27%	33%	40%	47%	53%	60%	67%	73%	80%	87%	93%	100%
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1200	0	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45
2400	0	3	6	9	12	15	18	21	25	28	31	34	37	40	43	46
3600	0	3	6	9	13	16	19	22	25	28	31	34	38	41	44	47
4800	0	3	6	10	13	16	19	22	26	29	32	35	38	42	45	48
6000	0	3	7	10	13	16	20	23	26	29	33	36	39	42	46	49
7200	0	3	7	10	13	17	20	23	27	30	33	37	40	43	47	50
8400	0	3	7	10	14	17	20	24	27	31	34	37	41	44	48	51
9600	0	3	7	10	14	17	21	24	28	31	35	38	42	45	49	52
10800	0	4	7	11	14	18	21	25	28	32	35	39	42	46	49	53
12000	0	4	7	11	14	18	22	25	29	32	36	40	43	47	50	54
13200	0	4	7	11	15	18	22	26	29	33	37	40	44	48	51	55
14400	0	4	7	11	15	19	22	26	30	34	37	41	45	49	52	56
15600	0	4	8	11	15	19	23	27	30	34	38	42	46	49	53	57
16800	0	4	8	12	15	19	23	27	31	35	39	43	46	50	54	58
18000	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60

applied design 5311 Western Ave., Suite 131 - Boulder, CO - 80301 - p.303-444-3125 - f.303-444-3126 - www.appliedesigncorp.com

Machine Readable																	
cylXDuration (pulses)		throttlePosition>>4 (8-bit /16)															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
crankVelocity >> 4 (pulses/16)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	4	8	12	16	21	25	29	33	37	42	46	50	54	59	63
	2	0	4	8	12	17	21	25	30	34	38	43	47	51	56	60	64
	3	0	4	8	13	17	22	26	30	35	39	44	48	52	57	61	66
	4	0	4	9	13	18	22	27	31	36	40	45	49	54	58	63	67
	5	0	4	9	13	18	22	27	32	36	41	45	50	55	59	64	68
	6	0	4	9	14	18	23	28	32	37	42	46	51	56	60	65	70
	7	0	4	9	14	19	23	28	33	38	43	47	52	57	62	66	71
	8	0	4	9	14	19	24	29	34	39	43	48	53	58	63	68	73
	9	0	4	9	14	19	24	29	34	39	44	49	54	59	64	69	74
	10	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75
	11	0	5	10	15	20	25	30	36	41	46	51	56	61	67	72	77
	12	0	5	10	15	21	26	31	36	42	47	52	57	63	68	73	78
	13	0	5	10	16	21	26	32	37	42	48	53	58	64	69	74	80
	14	0	5	10	16	21	27	32	38	43	48	54	59	65	70	76	81
	15	0	5	11	16	22	28	33	39	45	50	56	61	67	73	78	84

EXPECTED PERFORMANCE

20% friction losses are common in internal combustion engines. That number should be decreased somewhat by not needing to compress gases. The total power output will be a function of the duty cycle which the electromagnets are active, minus losses. Currently, the electromagnets are being fed 120 volts with an internal resistance of 9.8 ohms. That means that the electromagnets will be consuming roughly 12 amps and be consuming 1440 watts when on. If the electromagnets are only active on the downstroke for a full 180 degrees that will be a 50% duty cycle per electromagnet * 2 electromagnets, or 1440 watts. If the electromagnets are only active for the first 45 degrees that's 45 degree stroke / 360 degree total * 1440 watts per electromagnet * 2 electromagnets, or 360 watts.

It will be very important for performance of the engine to maximize the duty cycle which the electromagnets are active, as well as the power going through the electromagnet. For frame of reference common power ratings for vehicles are:

- Pedal assist bicycle: 250w – 1000w (1kw)
- Electric scooter: 1.5kw – 3kw
- Small electric motorcycle (125cc equivalent): 5kw – 7 kw
- Medium electric motorcycle (250cc equivalent): 10kw – 20 kw
- Zero SR-F electric motorcycle: 80kw
- Tesla Model 3 Standard: 200kw
- Tesla Model 3 Performance: 350kw

applied design 5311 Western Ave., Suite 131 - Boulder, CO - 80301 - p.303-444-3125 - f.303-444-3126 - www.appliedesigncorp.com

TERMINOLOGY

Because this study is focused on estimating the size, weight, and cost of the required battery pack for the product, electrical calculations and terminology will be thrown around extensively. Below is a brief description of the terminology and shallow theory pertaining to this study.

- **Watt (W)** – is a measurement of power, a typical heating pad uses 75 W of power, a typical automobile engine/motor produces 200,000 W (200 kW), a standard cell phone charger is 5-15 W. Watts can be derived by multiplying the voltage (V) and amperage (A) moving through it. A 3.7 V battery discharging at 2 A is providing 7.4 W, a 75 W light bulb plugged into a 110 V wall socket is pulling 0.68 A.
- **Volts (V)** – is (roughly speaking) the strength of electricity. Inside of your phone is a single 3.7 V battery, the wall produces 110 V, the outlet in a normal car is 12.6 V.
- **Amps (A)** – is a measure of electrons moving through something (typically measured in a copper wire). Old cell phones charged at 0.5 A over USB, the new USB-C standard can charge at 3 A, when you start your car it typically requires over 75 A.
- **Watt-hours (Wh)** – is a measure of the total amount of energy stored or used. It can be thought of has the number of hours something drawing 1 W can run for, or how many watts can be provided over an hour before that amount of energy is consumed. For example, a 75 W heated pad running for 10 hours will use 750 Wh of energy. A typical cell phone typically has a 7.4 Wh hour battery. Sometimes the watt-hour measurement of a battery has to be calculated by its multiplying its voltage and amp-hour rating (similar to how watts are derived from volts and amps), for example a sometimes a power bank says it has 5 Ah of capacity with a 3.7 V lithium batteries inside of it, it therefore has 18.5 Wh of capacity.
- **Amp-hours (Ah)** – is an indirect measure of the energy stored or being used by a device. It is a useful metric when every device being considered is at the same voltage, for example when pairing a cell phone with a portable power bank, because the batteries are all 3.7 lithium-ion batteries you know that 6 Ah (sometimes written as 6000 mAh) battery pack will recharge a 2 Ah cell phone 3 times. And if the battery pack can supply 2 amps of charging, it will take 1 hour to recharge a dead cell phone (it actually takes longer because of the charge curve of lithium batteries, but is outside the scope of this study).
- **Duty-cycle (%)** – Is a way of measuring the amount of a rated power being consumed by a device switching on and off (or varying the power levels directly), for example a 75 W heated mat running at 100% duty cycle is consuming 75 W of power. If the duty cycle is lowered to 50% it cycles it's 75 W element on and off so it's on half the time and off half the time, lowering the effective power consumption from 75 W to 37.5 W. 10% 7.5 W, 0% 0 W

ASSUMPTIONS

A number of assumptions must be made to constrain the calculations to a reasonable complexity. Having worked in electronics industry for many years, the following are good rules of thumb of parameters that can be expected in industry at a reasonable cost.

- **18650 Battery Cells** – these are the most common type of battery used in portable power applications of a reasonable size.
    - **3.7 V voltage (nominal)** – 4.2 V max, 3.3 V min
    - **3.350 Ah capacity**
    - **12.4 Wh capacity**
    - **18 mm x 65 mm cell dimensions** – dia x length, multiply by 1.25 to get pack level
    - **50 g cell weight** – multiply by 1.1 to get pack level
    - **$3.50 cell cost** – medium qty (1000X>) multiply by 1.15 to get pack level

AWG	Max amp	Ohms/m	Wire OD
18	16.00	0.019	1.062
20	11.00	0.030	0.848
22	7.00	0.048	0.678
24	3.50	0.077	0.541
26	2.20	0.122	0.432
28	1.40	0.195	0.345
30	0.86	0.310	0.277
32	0.53	0.492	0.224
34	0.33	0.783	0.178
36	0.21	1.244	0.142
38	0.13	1.979	0.114
40	0.09	3.147	0.089

applied design 5311 Western Ave., Suite 131 - Boulder, CO - 80301 - p.303-444-3125 - f.303-444-3126 - www.applieddesigncorp.com