AN4449
Application note

Buck-boost converter using the STM32F334 Discovery kit

Introduction

This application note describes the buck-boost DC/DC converter included in the STM32F334 Discovery kit (32F3348DISCOVERY), a low-cost and easy-to-use development kit to quickly evaluate and start application development with microcontrollers of the STM32F3 series.

The STM32F334xx ARM® Cortex®-M4 microcontrollers, combining high integration and performance, have been designed for digital power conversion applications and this buck-boost DC/DC converter illustrates how they can efficiently control an H-bridge topology converter. This application note demonstrates how the STM32F334xx products meet the needs of this function thanks to their embedded high-resolution timer (HRTIM) and the settings flexibility adapted for such switching-based converter application.

This demonstration example needs an external power supply that will be connected independently from the mini-B USB cable connected to the host PC.

The firmware associated to this example needs to be programmed into your STM32F334 Discovery kit prior to the demonstration.

Reference documents:

- UM1733: Getting started with STM32F334 Discovery kit;
- DB2343: Discovery kit for STM32F334 microcontrollers;
- UM1735: Discovery kit for STM32F3 series with STM32F334C8 MCU User Manual.

These documents are available on STMicroelectronics web site (http://www.st.com).

Before installing and using the product, please accept the Evaluation Product License Agreement from http://www.st.com/epla. For more information on the STM32F334 Discovery board and for demonstration software, visit www.st.com/stm32f3discovery.

Contents

List of tables

List of figures

1 Application description

1.1 Required hardware

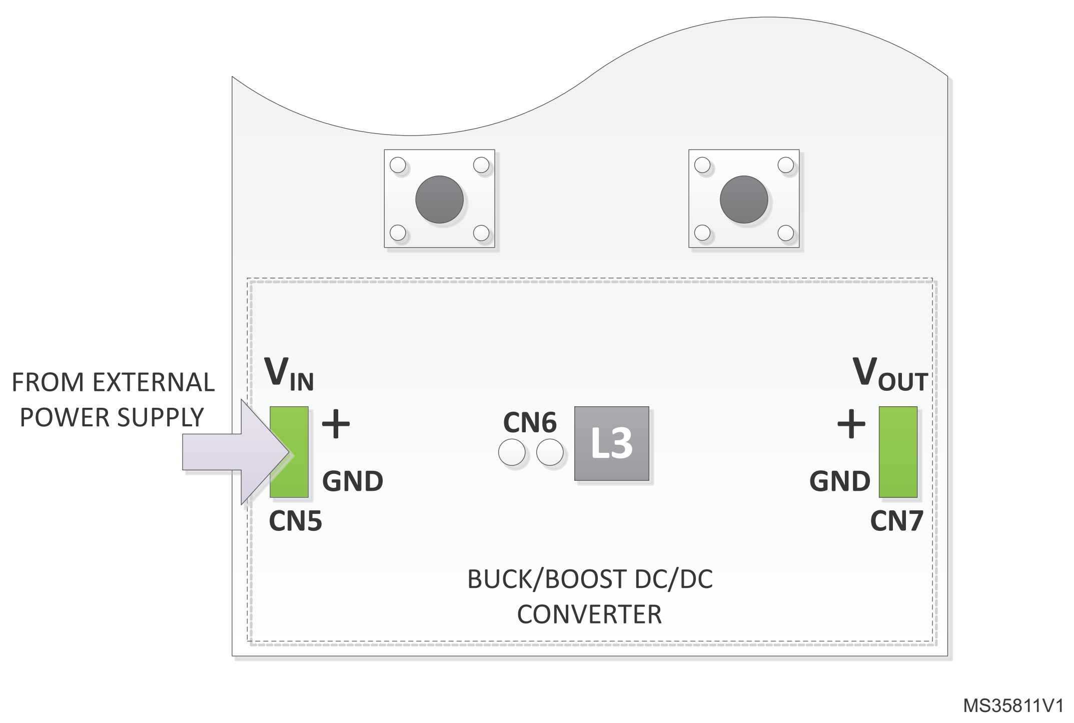
This application uses STM32F334 Discovery kit on-board buck-boost DC/DC converter and 4 signal LEDs (LD3 to LD6). An external DC power supply (0-15 V_{DC} 1 A max.) is required for the demonstration.

The external power supply will be connected to CN5 (V_{IN}) connector, while the V_{OUT} signal will be present on CN7.

The user will manage externally the current limitation to approximately 500 mA even if an internal protection against overload is included in this firmware demonstration.

Caution: Please comply with V_{IN} polarity when connecting the power supply as well for V_{OUT} when connecting an external load or metering tools, as shown in *Figure 1*.

Figure 1. Connecting the external power supply



Note: This example and its hardware are totally independent from high brightness LED features described in other documents.

1.2 Hardware settings of the STM32F334 Discovery kit

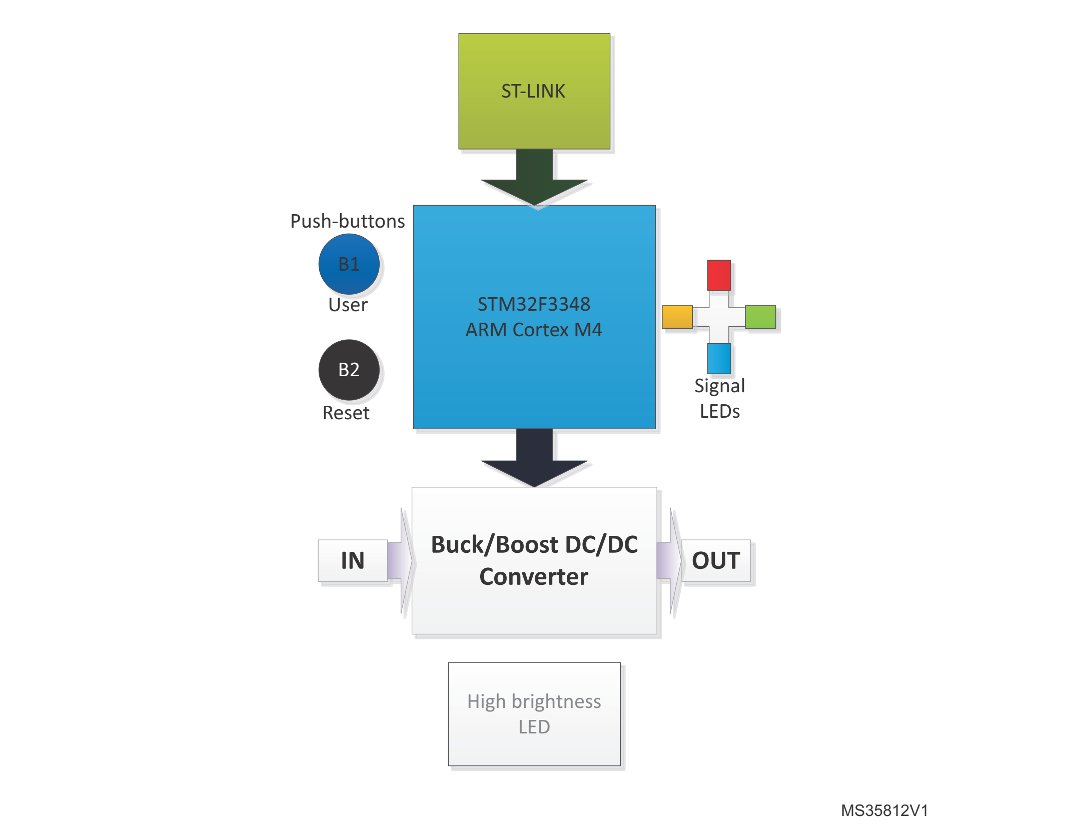
The SBx solder bridges should be set in their initial factory configuration. The user will possibly connect an external current probe on CN6 pads to observe the inductor (L3) current during operation. In that case, SB22 must be removed on the board bottom side.

Refer to STM32F334 Discovery kit user manual (UM1735) for further information related to hardware.

1.3 Application schematics

Figure 2 shows the description of STM32F334 Discovery kit hardware.

Figure 2. Overview of STM32F334 Discovery kit



1.4 Application principles

1.4.1 Overview

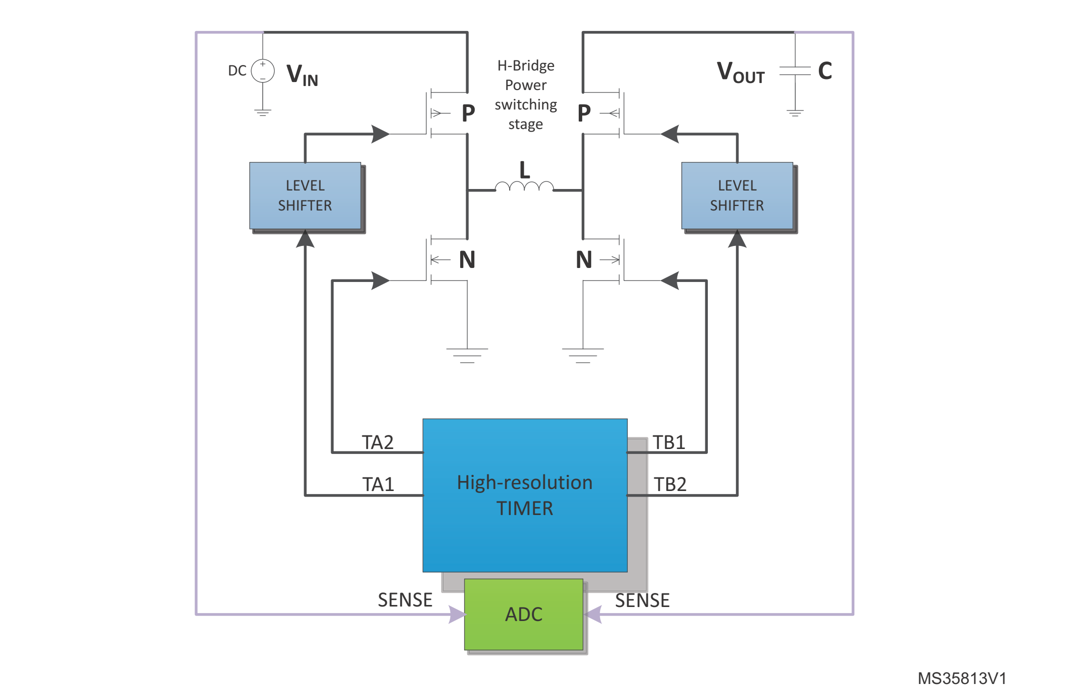
The STM32F334 Discovery embeds a buck-boost DC/DC converter that allows efficient conversion of a DC voltage from one level to another.

The user can apply an input DC voltage from a range of 3 to 15 V maximum and use the converter to get this voltage converted into a DC voltage ranging from 3 to 15 V maximum. The V_{OUT} target is set prior with a target constant defined into the firmware.

Due to the inductor size and the input/output varying conditions, the allowed output current is given for 0.5 A typical. The input voltage can be converted to a lower voltage level using a step-down mode, or converted to a higher level using a step-up mode. These 2 modes are supported with a single H-bridge converter using a non-inverting buck-boost topology.

The STM32F334 microcontroller is interfaced to the power switches with a minimum hardware, some acting as level shifters especially to adapt the low-level voltage of microcontroller supply to the power stages connected to the external source. The buck-boost converter topology is sketched in *Figure 3*.

Figure 3. STM32F334 Discovery Buck/Boost converter topology



As shown above, the high-resolution timer controls the N and P MOS switches of H-bridge converter. The P-MOSFETs are not directly connected to the high-resolution timer but interfaced with level shifters, while the logic level N-MOSFETs are directly controlled by the MCU. These level shifters are made up of a bipolar totem-pole driver. Please refer to UM1735 for further details on converter schematics.

The buck-leg (left side of H-bridge) is driven by TA1 (for the PMOS) and TA2 (for the NMOS) signals from HRTIM (high-resolution timer) and the boost-leg (right side of H-bridge) is driven by TB1 (for the NMOS) and TB2 (for the PMOS) signals from HRTIM. This converter structure uses 4 MOSFET switches to allow synchronous rectification that improves converter efficiency and thermal performance. This consists to replace the rectification diodes commonly used in buck or boost converter stages by MOSFET transistors ensuring the current rectification and providing better efficiency in low-output-voltage and high-current power supplies. The conduction losses in the freewheeling diodes can represent a significant part of the power losses of switching converters.

The internal HRTIM PWM switching frequency is 250 kHz. Both V_{IN} and V_{OUT} signals are sensed by the ADC peripheral and the varying error vs. the V_{OUT} target is evaluated.

This mode is called voltage mode control as the output regulation is based on voltage measurement. The PWM duty is then computed internally by a proportional integral controller (PI) based on ADC inputs.

The input or output currents are not sensed by any hardware current sensing method (series sense resistor, MOSFET current sensing, current transformers, etc...) but evaluated by a software solution described later in this document.

Caution: For safety reasons, users must not modify the current design and must comply with firmware operating limitations. Moreover, USB voltage source is not recommended as an alternate power supply source for V_{IN} as this can damage the board or the host USB PC port if a high current is forced in the converter.

1.4.2 Non-inverting buck/boost converter basics

Step-down operation (buck mode)

This mode operates when V_{IN} >> V_{OUT} target and when the converter lowers the input voltage level.

As shown in *Figure 4*, TA1 and TA2 are two PWM signals generated by the HRTIM and act together as complementary signals. As evidenced by the converter topology, two transistors from a same leg cannot be switched at the same time (thus risking to create a short-circuit on the power line). Therefore Q1 and Q2 cannot be turned ON together and must be managed one by one with the HRTIM peripheral by inserting dead-time periods between conductions of Q1 and Q2 as shown in *Figure 5*. When Q1 is closed, the inductor charge phase is observed. When Q2 is closed at its turn, the inductor discharge phase can start. On the right side of the H-bridge, Q4 is maintained closed and Q3 open during all buck operating phase.

Note: In figures 4 and 5 the same color scheme has been used to indicate current flows.

Figure 4. Step-down operation (buck mode)

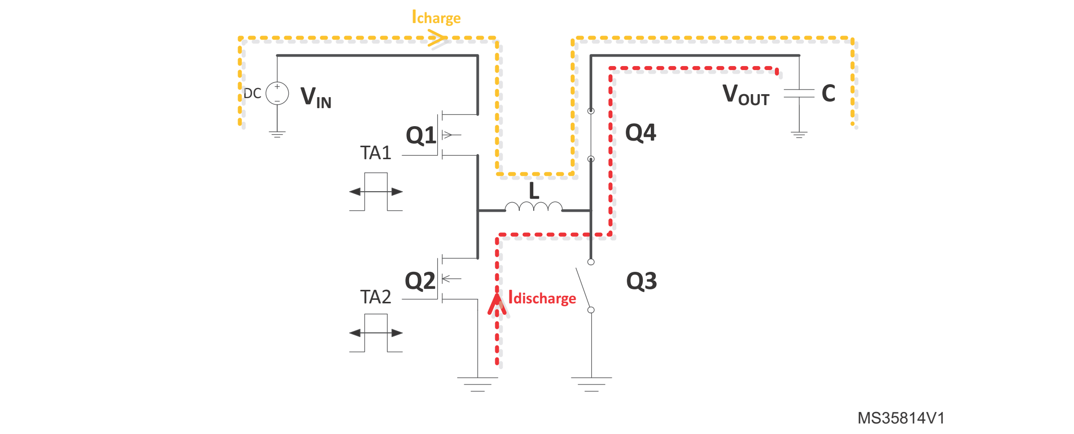
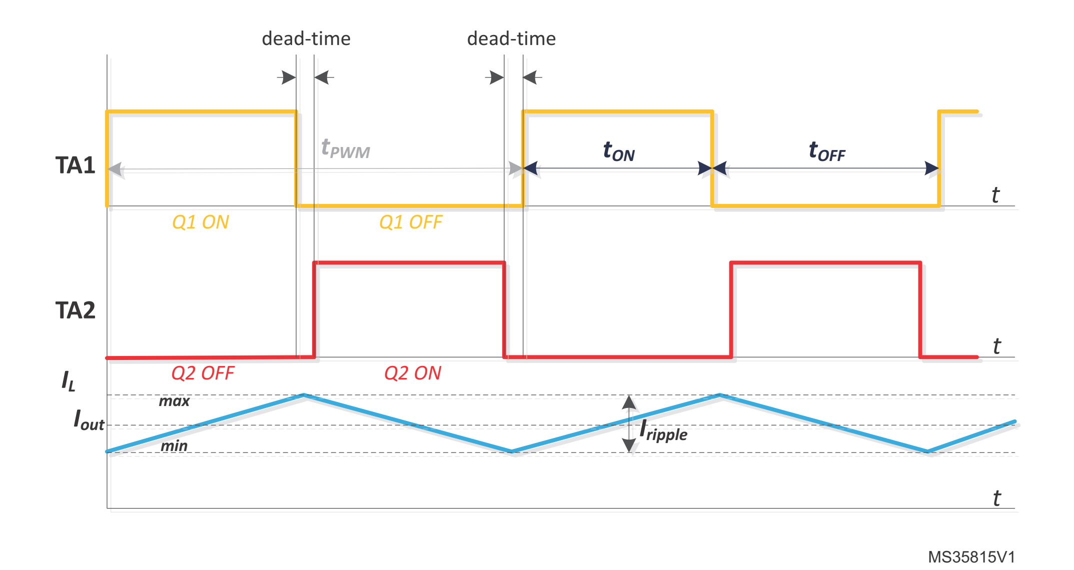


Figure 5. Step-down operation signals (buck mode)



During charge phase, Q1 is closed and the V_{IN} voltage is applied to the inductor. The current flows through the inductor and the output capacitor is charged through the ground. If the voltage across Q1 and Q4 are ignored (ideal switches), the relation between V_{IN}, V_{OUT}, L and the current I_L flowing through L can be expressed by the formula below:

$$V_{in} - V_{out} = L\frac{di_L}{dt}$$

During discharge phase, Q1 is open and the current continues to flow through Q2 (now closed) and through the inductor. In this condition and with Q2 and Q4 drop voltages ignored as well, the formula between the current, L and V_{OUT} becomes:

$$-V_{out} = L\frac{di_L}{dt}$$

If the HRTIM PWM switching period is T_{PWM} (in seconds) then the charge phase t_{ON} and the discharge phase t_{OFF} are linked together by the formula:

$$t_{ON} + t_{OFF} = T_{PWM}$$

The PWM duty cycle is the ratio of time when Q1 is turned ON over the total T_{PWM} period. D is always included between 0 and 1. It can be expressed as:

$$D = \frac{t_{ON}}{T_{PWM}}$$

When Q1 is turned OFF, the OFF time is equivalent to:

$$t_{OFF} = (1 - D) \times T_{PWM}$$

Considering that the average voltage across the inductor L is zero during a whole PWM period and to prevent any saturation into the coil, the relation between V_{IN} and V_{OUT} for the step-down converter (buck mode) is:

$$D \times (Vin - Vout) + (1 - D) \times (-Vout) = 0$$

$$D \times Vin - D \times Vout - Vout + D \times Vout = 0$$

This sums up to:

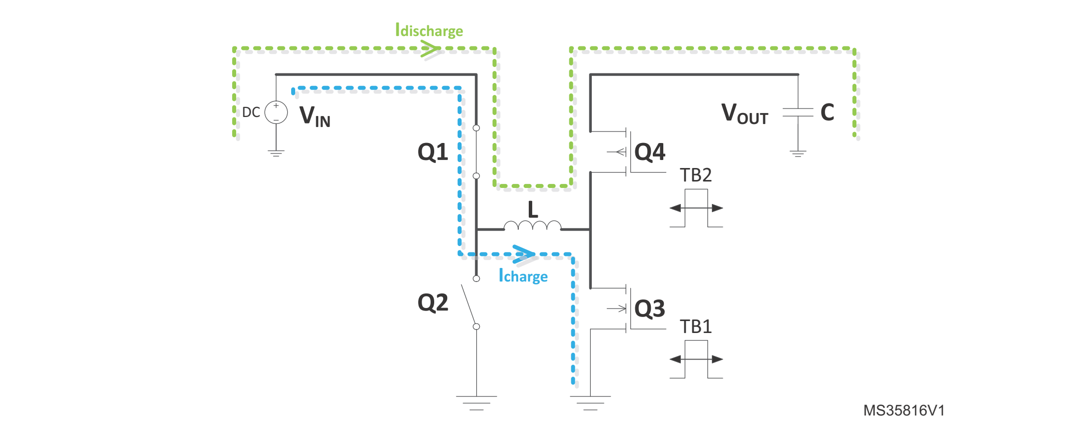
$$\frac{V_{out}}{V_{in}} = D$$

Step-up operation (boost mode)

This mode operates when $V_{IN} << V_{OUT}$ target and when the converter raises the input voltage level. As shown in *Figure 6*, TB1 and TB2 are two PWM signals generated by the HRTIM and are working together as complementary signals. As it was the case in the buck mode, and for the same H-bridge leg, Q3 and Q4 cannot be turned ON at the same time, as shown in *Figure 7*. When Q3 is closed, the inductor charge phase takes place. When Q4 is closed, the inductor discharge phase starts. On the left side of the H-bridge, Q1 is maintained closed and Q2 open during all boost operating phase.

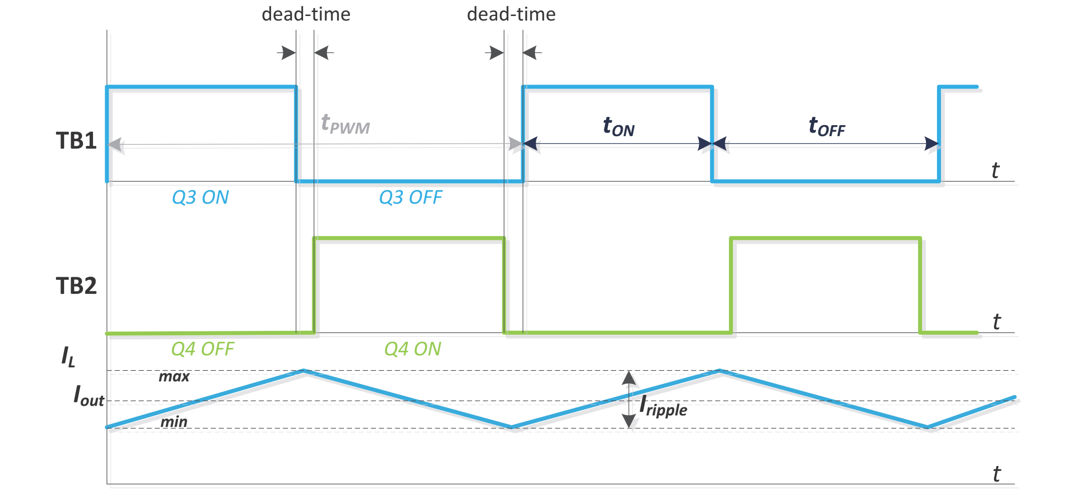
Note: In figures 6 and 7 the same color scheme has been used to indicate current flows.

Figure 6. Step-up operation (boost mode)



MS35816V1

Figure 7. Step-up operation signals (boost mode)



MS35817V1

During charge phase, Q3 is closed and the V_{IN} voltage is applied on the inductor through the ground. If the voltages across Q1 and Q3 are ignored (ideal switches), the relation between V_{IN}, L and the current I_L flowing through L can be expressed by the formula below:

$$V_{in} = L\frac{di_L}{dt}$$

During discharge phase, Q3 is open and the current continues to flow through the inductor and Q4 (now closed) and finally through the output capacitor. In this condition and with Q1 and Q4 drop voltages ignored as well, the formula between V_{IN}, the current I_L, L and V_{OUT} becomes:

$$V_{in} - V_{out} = L\frac{di_L}{dt}$$

For the charge phase t_{ON} and the discharge phase t_{OFF} the same formula can be applied:

$$t_{ON} + t_{OFF} = T_{PWM}$$

The PWM duty cycle is the ratio of time when Q3 is turned ON over the total T_{PWM} period. It can be expressed with D always included between 0 and 1as:

$$D = \frac{t_{ON}}{T_{PWM}}$$

When Q3 is turned OFF, the OFF time is equivalent to:

$$t_{OFF} = (1 - D) \times T_{PWM}$$

For the boost mode, the following formula can be applied:

$$D \times Vin + (1 - D) \times (Vin - Vout) = 0$$

$$D \times Vin + Vin - Vout - D \times Vin + D \times Vout = 0$$

$$Vin = Vout(1 - D)$$

That finally gives:

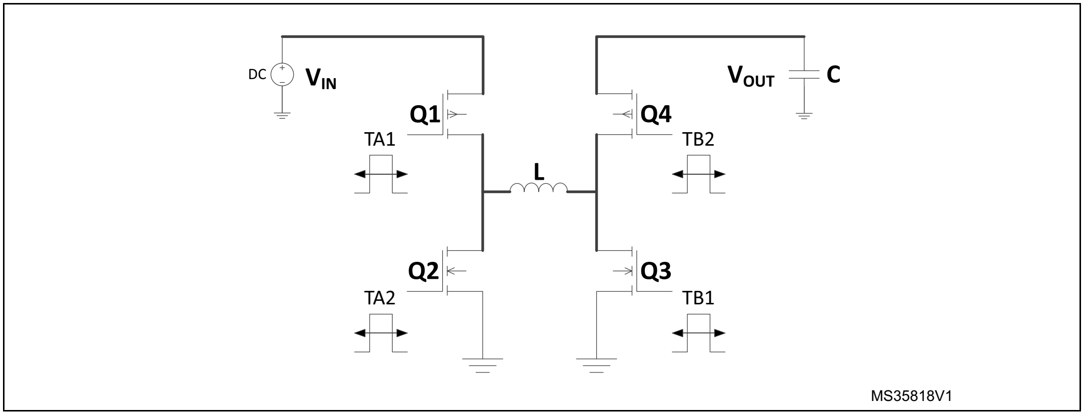
$$\frac{V_{out}}{V_{in}} = \frac{1}{1 - D}$$

Buck-boost operation (mixed mode)

This mode occurs when V_{IN} is near V_{OUT} (V_{IN} ~ V_{OUT}) and none of the buck or boost modes can independently achieve V_{OUT} target regulation. It uses both buck and boost sides of the H-bridge converter and the 4 HRTIM outputs TA1, TA2, TB1 and TB2 generate PWM signals where TA1 and TB1 are still complementary respectively to TA2 and TB2.

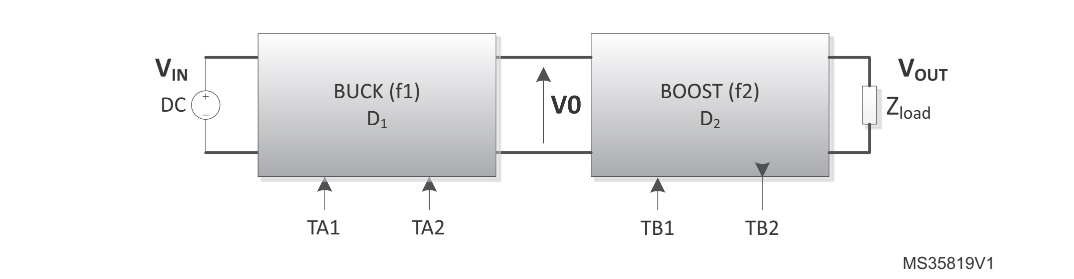
In this application example, during buck-boost operation, the buck duty cycle is maintained fixed while the boost duty cycle is variable according to V_{IN} and V_{OUT} conditions.

Figure 8. Buck-boost operation (mixed mode)



In the buck-boost mode, both operations are cascaded. Considering the buck mode as a transfer function f1 with D1 as an input parameter and the boost mode as a transfer function f2 with D2 as an input parameter, the equivalent scheme shown in *Figure 9* can be applied:

Figure 9. Buck-boost cascaded transfer functions



Now applying the consecutive equations for buck and boost converters:

$$V0 = f1(D1) \times Vin$$

and

$$Vout = f2(D2) \times V0$$

then the global transfer function f is:

$$\frac{Vout}{Vin} = f(D1, D2) = f1(D1) \times f2(D2)$$

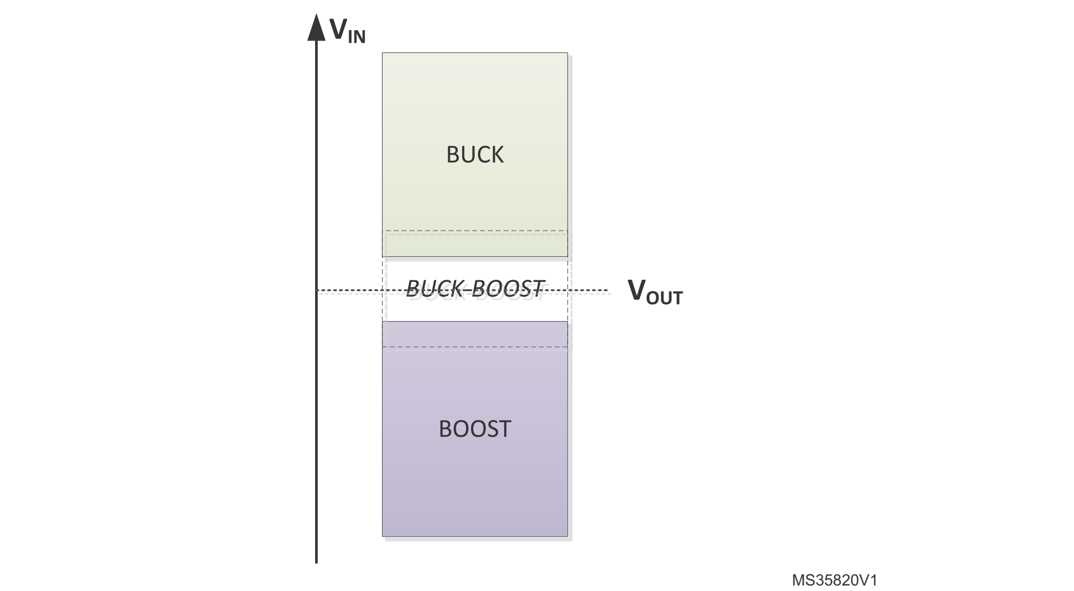
As the transfer functions based on the converter duty cycles for buck and boost have been mentioned above in this document, this finally gives for the buck-boost converter mode:

$$\frac{Vout}{Vin} = D1 \times \frac{1}{1 - D2} = \frac{D1}{1 - D2}$$

As an example, setting D1 buck duty cycle to 0.8 (fixed), with a variation of D2 boost duty cycle between 0.05 and 0.45, allows to have a V_{OUT} / V_{IN} ratio varying from 0.85 to 1.45, thus covering the entire range of V_{OUT} close to V_{IN}.

The purpose is to have 3 operating modes that are slightly overlapped between adjacent ranges, as graphically shown in *Figure 10*.

Figure 10. Operating modes according to V_{IN} level



This overlap prevents from sporadic switching from one mode to another especially when the converter operation is located at the threshold of the maximum or minimum operating duty for the current mode. It acts like a hysteresis and improves application stability.

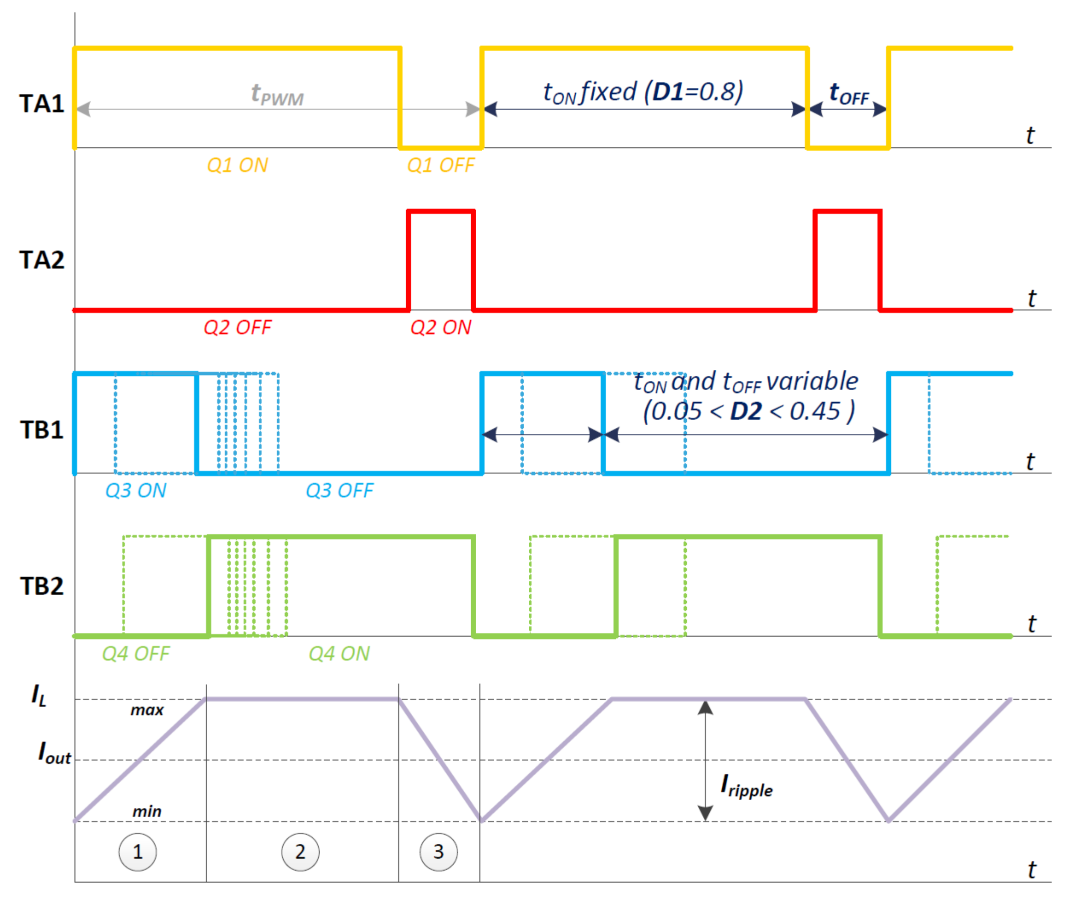
Consequently, if a V_{OUT} target is set, the converter can operate in the 3 areas, for the cases $V_{IN} << V_{OUT}$, $V_{IN} \sim V_{OUT}$, or $V_{IN} >> V_{OUT}$.

Concerning the buck-boost operating signals, the buck operation is achieved with a fixed PWM waveform on TA1 (TA2 as complementary) and the boost one with a variable PWM duty cycle on TB1 (TB2 as complementary) as shown in *Figure 11*, where the same color scheme of figures *4* to *7* has been used for currents.

In this example, the PWM duty cycle D1 applied on buck side is fixed to 0.8 and the PWM duty cycle D2 applied on the boost side may vary from 0.05 up to 0.45. Theoretically, if V_{IN} = V_{OUT}, D1 and D2 should be complementary, as for instance, if D1 is fixed to 0.8, then D2 should be 0.2 referring to the relation between V_{IN}, V_{OUT}, D1 and D2 seen above. In real application case, D2 needs some margins to introduce more flexibility between the conversion modes and ensure a minimum hysteresis over operating ranges.

The I_L current waveform can be broken down into 3 regions, as shown in *Figure 11*.

Figure 11. Buck-boost operation signals (mixed mode)



The first phase is equivalent to the boost time when phases 2 and 3 correspond to the buck time. This curve is represented when V_{IN} is very close to V_{OUT} due to the horizontal line observed in section 2. If V_{IN} is slightly lower than V_{OUT}, the line slope located in section 2 becomes negative and the current decreases through the inductor during this time slot. On the opposite, if V_{IN} is slightly higher than V_{OUT}, the line slope located in section 2 is positive and the current through the inductor increases for this same time slot.

Equations previously defined for buck and boost modes can be applied for the required period.

1.4.3 Sizing the inductor of the buck-boost converter

Objectives of this buck-boost converter are to have an input voltage range from 3 V_{DC} to 15 V_{DC} maximum and the same parameters for the output voltage. For the converter, the typical output current is 500 mA. The HRTIM switching frequency is selected to have a PWM frequency of 250 kHz. A relevant estimation for the inductor ripple current is 30% typical of the output current value.

For the buck mode, with a maximum of 15 V_{DC} in input and a minimum of 3 V_{DC} on the output, the value of theoretical duty D is 3/15 = 0.2.

Based on these parameters and from the initial equation:

$$V_L = L\frac{di_L}{dt}$$

The inductor L estimation value can be calculated with the formula below:

$$L_{\min} > \frac{(Vin - Vout)}{Fpwm} \times D \times \frac{1}{Iripple}$$

$$L_{\min} > \frac{(15-3)}{250000} \times 0.2 \times \frac{1}{0.5 \times 0.3} = 64\mu H$$

For the boost mode, the input and output ranges are similar and the formula is now:

$$L_{\min} > \frac{Vin^2}{Fpwm} \times \frac{(Vout - Vin)}{Vout^2} \times \frac{1}{Iripple}$$

The chosen value is L = 82 µH with a resistance r_L of 460 mΩ typical, and typical saturation current of 1.1 A.

1.4.4 Setting the STM32F334 High-resolution timer

This section describes how the STM32F334 high-resolution timer is set for this application example. As described above, the N and P MOSFETs of the H-bridge converter are each ones assigned to one of the HRTIM outputs. For the buck right side, there are TA1 and TA2 PWM waveforms connected to the buck-leg and for the boost left side, TB1 and TB2 PWM waveforms connected to the boost-leg. TA1 and TA2, as well as TB1 and TB2 act as complementary signals.

There are 4 modes that control the buck boost converter, as described in *Table 1*.

Table 1. HRTIM output signals through operating modes

HRTIM output	Converter mode			
	IDLE	BUCK	MIXED	BOOST
TA1	Low	PWMA	PWMA	High
TA2	Low	/PWMA	/PWMA	Low
TB1	Low	Low	PWMB	PWMB
TB2	Low	High	/PWMB	/PWMB

IDLE mode is a waiting mode when only timers A and B have been started but none of the outputs has been set yet.

As soon as outputs from Timer A or Timer B are controlled, there is one state configured for the timer period event and another one configured for a compare event (or a few ones). The PWM period is selected to 4µs (f_{PWM} = 250 kHz). Each 4 µs, the timer period event occurs. Since both timer A and timer B are synchronized together, the period event is common for the 2 timers. To activate the related outputs to timer A and B, one (or more) compare events is (are) created, as shown in *Table 2*.

Table 2. Event settings through operating modes

	Converter mode									
	ALL	BUCK			MIXED			BOOST		
	Period	CMP1	CMP2	CMP4	CMP1	CMP2	CMP4	CMP1	CMP2	CMP4
Timer A	4 µs	DutyA	-	ADC trigger	DutyA		ADC trigger	-	-	ADC trigger
Timer B	4 µs	-	-	-	-	DutyB Mixed	-	DutyB	-	-

Sliding ADC trigger event

Switching converters generate noise on the power lines due to the fast switch of the power MOSFETs. Nevertheless, the application has to measure regularly the values of V_{IN} and V_{OUT} voltages during operation and potentially during switching time periods.

According to the different operating modes, buck, mixed and boost modes, there are various switching events on the timers outputs that can degrade the ADC measure when performed at a regular time without looking at conditions on HRTIM outputs. For instance, if the ADC measure is defined at the half of the PWM period while the duty cycle of an HRTIM output signal reaches 0.5, both ADC and switching event are performed at the same time. To ensure that the ADC measure is not polluted by the noise during a MOSFET transition, the ADC trigger event has to be determined according to the current duty cycle value and also depending on the various switching events due to the considered mode.

To improve V_{IN} and V_{OUT} ADC measurement accuracy and to carry out the sampling in a non-transition period, the ADC trigger event is adapted every time the duty cycle changes.

There are 2 conditions that are checked before applying the ADC trigger event time:

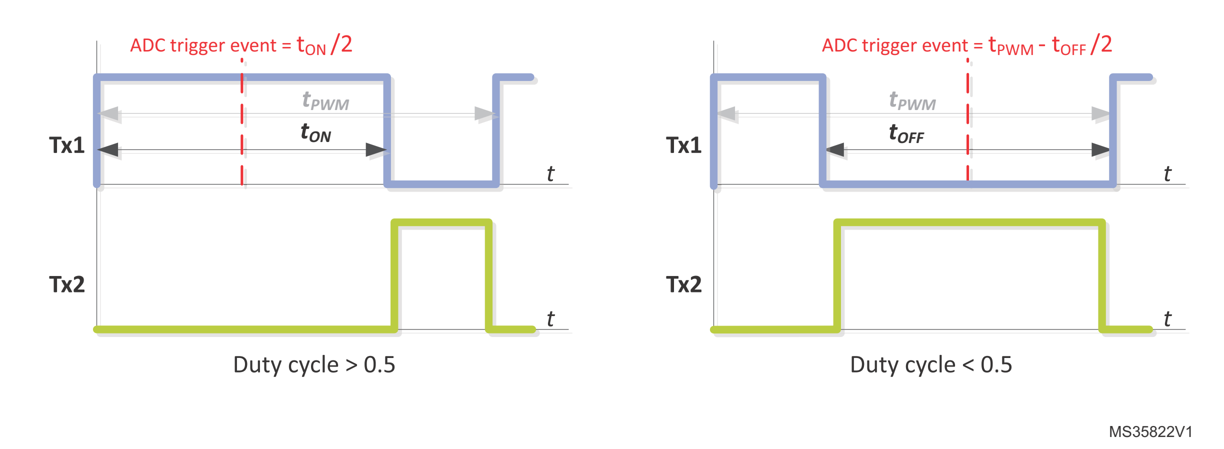
1. Buck or Boost mode operation:

   In this case, only one group of complementary outputs is active (TAx or TBx). The time value of the ADC trigger event depends essentially on the value of the running duty cycle. The ADC trigger time is assigned on the opposite sector of the signal transition and its event also slides with the value of the duty cycle. It depends upon t_{ON} or t_{OFF} value, therefore the ADC trigger position varies with duty cycle as shown in *Figure 12*.

2. Mixed (buck-boost) mode operation:

   In this case, both groups TAx and TBx are active (see *Figure 11*). As the buck mode is set with a fixed duty cycle of 0.8 and the boost mode duty cycle cannot exceed 0.45, therefore the ADC trigger time event is set to a fixed value corresponding to a free measuring window located in between. The value of ADC trigger time is typically 60% of the overall PWM period.

Figure 12. ADC trigger event configuration in buck and boost modes



1.4.5 Software overload protection

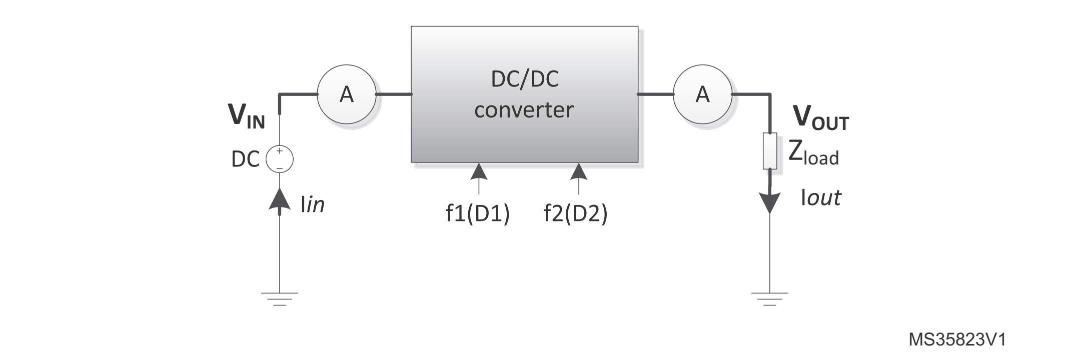
Buck-boost converters generally use different methods to evaluate the load current sourced by the converter. There are several means but the most common method is to sense the load current with a current sense resistor inserted serially into the output circuitry. This allows controlling the maximum inductor current and to prevent any overload conditions. The drawback is to introduce power losses caused by the inserted resistor and resulting in a lower converter efficiency. Even if there are lossless current-sensing methods, here a different approach has been used for this application example.

As described above in step-up or step-down operations in buck-boost basics chapter, the theoretical duty cycle is known for a given operating mode (buck, boost, etc...) and can be calculated by the V_{IN} / V_{OUT} ratio applied onto the converter. As for instance, a 10 V_{DC} input voltage can be converted into 5 V_{DC} output voltage with a duty cycle equal to 0.5 (buck mode). This basic assumption does not take into account the DC/DC converter efficiency for a given output load current. The relation for buck mode is actually where η is the efficiency factor:

$$\frac{V_{out}}{V_{in}} \times \eta = D$$

Therefore, the application cannot anticipate itself the approximate value of the output current based on the running mode and the duty cycle applied during operation.The idea is to have this information available and stored into the microcontroller memory by anticipating the efficiency based on the real case operation. For this purpose, the following configuration is set as in *Figure 13*.

Figure 13. DC/DC converter characterization



This DC/DC converter has been designed to have a typical current of 0.5 A for both I_{in} and I_{out}.

The idea is to characterize the application in the different operating modes according to various V_{IN} / V_{OUT} ratios: different (V_{IN}, V_{OUT}) pairs are applied to the converter, and two ammeters inserted on input and output lines. For each pair of voltage values, the variable Z_{load} impedance is adjusted so that I_{in} and I_{out} are always within the desired range.

As an example, an input voltage of 10 V_{DC} is applied on V_{IN}, and V_{OUT} target is set to 5 V_{DC}. The impedance load is tuned to obtain I_{out} = 0.55 A. The duty cycle is recorded for the operating point (V_{IN} = 10 V, V_{OUT} = 5 V). Similarly, for another (V_{IN}, V_{OUT}) pair, this time in boost mode, the impedance load is tuned to have I_{in} not exceeding 0.55 A. Thus, a raw table containing duty cycles D1 or D2 is reported for each (V_{IN}, V_{OUT}) pair. This allows determining the functional mapping for the different modes.

The input range V_{IN} is included between 3 to 15 V_{DC} as well as for the output range. For each (V_{IN}, V_{OUT}) pair, the converter applies the function f1(D1) or f2(D2) or f(D1,D2) = f1(D1) X f2(D2) according to the V_{IN} / V_{OUT} ratio and the selected operating mode (buck, boost or mixed).

The results are represented graphically in *Table 3*, *Table 4* and *Table 5*, related to the three operating modes. In each table, cells in green allow to easily identify the operating area for the concerned mode (for readability purposes, duty cycles values have been removed). In these tables V_{IN} and V_{OUT} are both expressed in Volts.

As an example, the duty cycles values Y0 to Y12 are obtained by f1 function:

$$y0 = NA; y1 = f1(4,3); y2 = f1(5,3); etc...y12 = f1(15,3)$$

Table 3. Collection table for buck mode

f1(D1)		V_{IN}												
		3	4	5	6	7	8	9	10	11	12	13	14	15
V_{OUT}	3	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12
	4			🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
	5					🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
	6						🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
	7							🟩	🟩	🟩	🟩	🟩	🟩	🟩
	8								🟩	🟩	🟩	🟩	🟩	🟩
	9									🟩	🟩	🟩	🟩	🟩
	10											🟩	🟩	🟩
	11												🟩	🟩
	12													🟩
	13													
	14													
	15													

The same duty cycle collections are gathered for boost and mixed modes (*Table 4* and *Table 5*, respectively). All these modes are indeed slightly overlapped together.

Table 4. Collection table for boost mode

f2(D2)		V_{IN}												
		3	4	5	6	7	8	9	10	11	12	13	14	15
V_{OUT}	3	🟩												
	4	🟩	🟩											
	5	🟩	🟩	🟩										
	6	🟩	🟩	🟩	🟩									
	7	🟩	🟩	🟩	🟩	🟩								
	8	🟩	🟩	🟩	🟩	🟩	🟩							
	9	🟩	🟩	🟩	🟩	🟩	🟩							
	10	🟩	🟩	🟩	🟩	🟩	🟩	🟩						
	11	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩					
	12	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩				
	13	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩			
	14	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩		
	15	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	

Table 5. Collection table for mixed mode

f(D1, D2)		V_{IN}												
		3	4	5	6	7	8	9	10	11	12	13	14	15
V_{OUT}	3		🟩											
	4			🟩										
	5				🟩									
	6				🟩	🟩	🟩							
	7					🟩	🟩	🟩						
	8						🟩	🟩	🟩					
	9							🟩	🟩	🟩	🟩			
	10								🟩	🟩	🟩	🟩		
	11								🟩	🟩	🟩	🟩	🟩	🟩
	12									🟩	🟩	🟩	🟩	🟩
	13										🟩	🟩	🟩	🟩
	14											🟩	🟩	🟩
	15												🟩	🟩

Going further, if the duty cycle values extracted from a same row table (Y0, Y1, Y2, etc... up to Y12) are represented into a graph, it can be observed a corresponding curve or line for any row of the table.

As shown in *Figure 14*, every buck function f1(D1) represents a curve for each V_{OUT} value chosen where other functions for respectively boost (*Figure 15*) and mixed (*Figure 16*) modes as f2(D2) and f(D1,D2) are equivalent to simple lines.

Figure 14. Duty cycle value vs. V_{IN} for a given V_{OUT} in buck mode

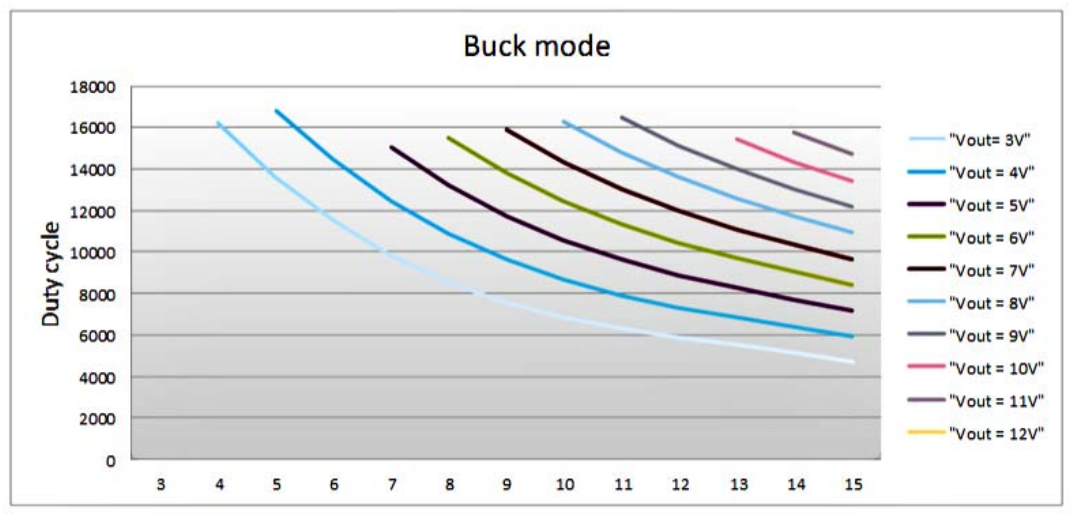


Figure 15. Duty cycle value vs. V_{IN} for a given V_{OUT} in boost mode

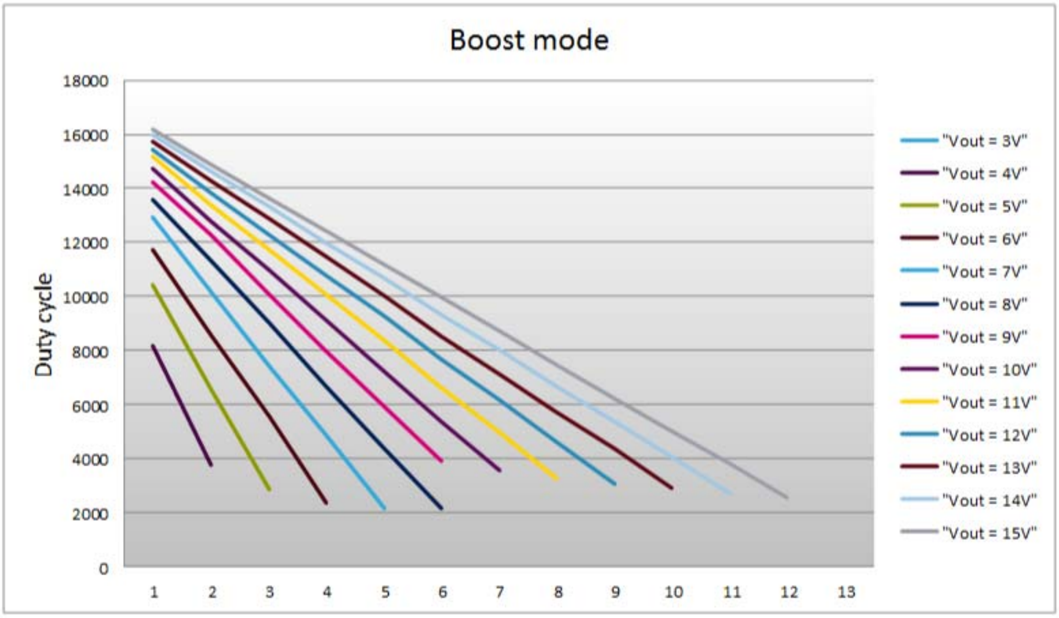
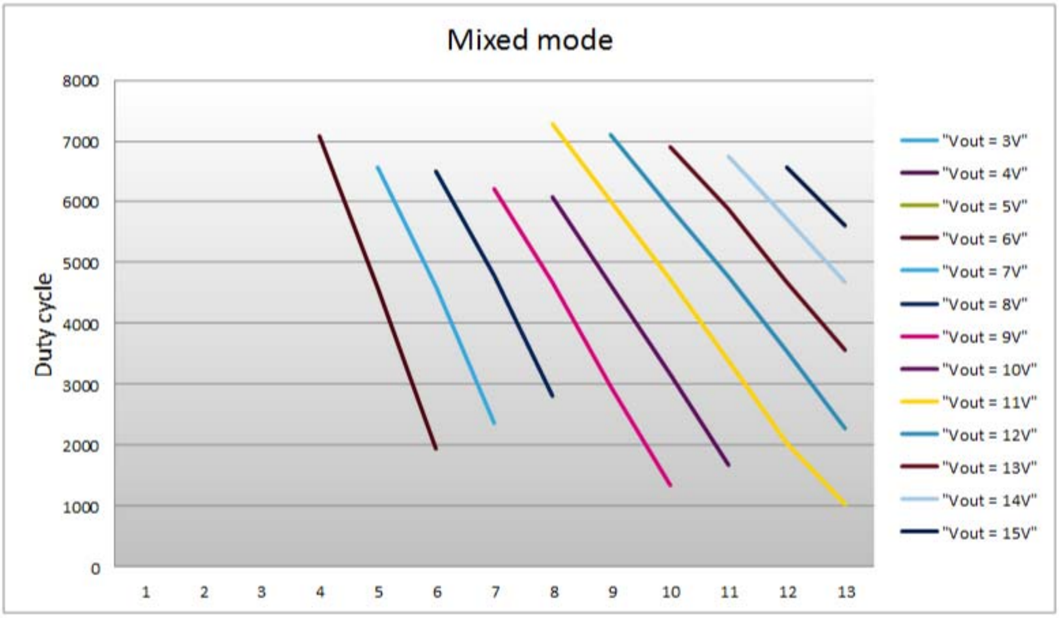


Figure 16. Duty cycle value vs. V_{IN} for a given V_{OUT} in mixed mode



The goal is to have all these data stored in three different tables, by keeping as information only the curve or line equations needed to calculate the corresponding duty cycle for the running (V_{IN}, V_{OUT}) pair. For the buck mode, the equations of these curves are computed with polynomial interpolation while for both boost and mixed modes only regression lines calculation is necessary.

To compute the polynomial interpolation based on Lagrange's interpolation, at least 4 distinct points are extracted from the raw table. This operation results in 4 polynomial factors from degree included between 0 and 3 that are combined to get the final equivalent function.

As for instance, 4 points coordinates are extracted from the first buck curve: p1(5, 13605); p2(8, 8521); p3(11, 6285); p4(15, 4731). Please notice that all duty values are expressed for a whole PWM period equal to 18432 units. This number corresponds to 4 µs multiplied by the timer frequency, equivalent to 144 MHz X 32. The corresponding duty cycle for 13605 is indeed 13605/18432 = 73%. After Lagrange's interpolation is computed, the equation of the interpolated curve is:

$$L(x) = \frac{-13513}{1260} \times x^3 + \frac{130918}{315} \times x^2 - \frac{7199839}{1260} \times x + \frac{695647}{21}$$

Finally, these factors are arranged and simplified to limit the amount of data stored in software tables with a slight loss of accuracy for the last constant term.

This gives:

$$L(x) = -10.725 \times x^3 + 415.612 \times x^2 - 5714.157 \times x + 33126.047$$

The data stored in table for the equation curve corresponding to V_{OUT} = 3 V for the buck mode will be: {10725; 415612; 5714157; 33126}. From these data, the curve equation can be retrieved easily and the last required operations will be performed by the MCU to calculate the corresponding duty from an input value of V_{IN}.

It's now possible to calculate any duty cycle for the buck mode for the V_{IN} range [3;15] V_{DC}.

Which would be the duty cycle for V_{IN} = 9 V to obtain V_{OUT} = 3 V in buck mode? This means for the MCU to extract the 4 factors and to calculate:

$$L(9) = -10.725 \times (9)^3 + 415.612 \times (9)^2 - 5714.157 \times 9 + 33126$$

This finally gives:

$$L(9) = 7544$$

Thus, if V_{IN} = 9 V and V_{OUT} = 3 V then 7544 (corresponds to 41%) represents the duty cycle register value that application shouldn't exceed to keep the output (or input) current below typical value. If the duty cycle order is set beyond, therefore the output current will raise over it. In that case, overload conditions on the converter are reached.

As well for boost and mixed modes, the regression line is computed based on the coordinates of all different points collected in the raw data tables.

In that case, only 2 factors a and b will result from the regression line such as:

$$f(x) = ax + b$$

All these data are stored in 3 tables, the first one for buck mode including the 4 polynomial factors (x^3 to x^0) for each of the 0 to 12 rows (3 to 15 V). The second and third one (respectively assigned to boost and mixed mode) contain the 2 regression line factors a and b for each of the 0 to 12 rows. If user has chosen an intermediate value for V_{OUT} (not an integer one, e.g. V_{OUT} = 5.7 V), the MCU computes the value for two adjacent curves (or lines) and performs a linear interpolation between these 2 curves. As an example, if the user has targeted V_{OUT} = 5.7 V, then the duty cycles from adjacent curves or lines for 5 and 6 V are computed with V_{IN} value and a linear interpolation is performed to find the nearest approximation of maximum duty cycle.

Basically, for a given mode, the application uses the corresponding equation to evaluate the maximum duty cycle admissible for the running operation. Subsequently, the value of the duty cycle applied into the converter is compared with the value of the computed duty cycle limit. If the duty cycle applied into the converter exceeds the value of the theoretical limit for a given time, then the converter is stopped and the overload is detected. As a drawback, there is no possibility to change the overload limit as the data entered into tables, then the interpolation functions, have been characterized for a defined current value.

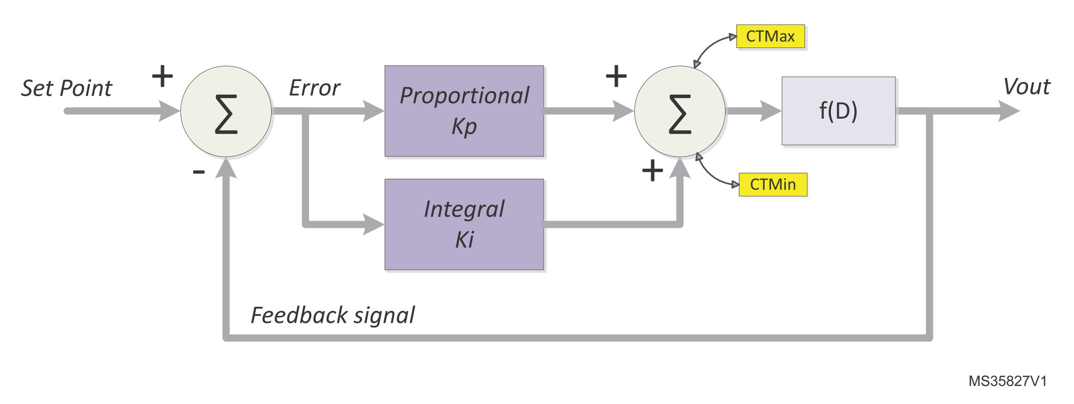
Nevertheless, this method does not use external hardware that would degrade converter efficiency such as current-sense resistors but can be easily used for current limitation purpose. This allows stopping the converter and anticipating any hardware damages if an external output load is forcing the converter beyond its capability.

1.4.6 PI software regulation

This DC/DC converter uses the measured and the target V_{OUT} voltages to set the converter duty cycle in the corresponding operating mode. As mentioned earlier, the ADC peripheral is regularly triggered by the HRTIM to sample V_{IN} and V_{OUT} values. Based on this information, a software based PI controller (proportional-integral) is used to apply the optimized value of duty cycle according to V_{OUT} target value.

As shown in *Figure 17*, the application set point which is the V_{OUT} target of the converter is compared with the V_{OUT} fed back by ADC conversion.

Figure 17. PI software controller



The error is computed by the difference between the target value and the ADC measured value. This error is then amplified with Kp and Ki gain parameters from respectively Proportional and Integral terms and summed together to set the new duty cycle order. Depending on the considered converter mode (buck, boost or mixed), the PI controller setting value is limited to the maximum or minimum duty cycle value allowed for the related mode by a saturation method. This saturation can take place for very low or very high values of duty cycle. Every time the saturation point is reached during a PI request, one of the two counters, CTMax or CTMin, respectively assigned to maximum and minimum duty cycle is incremented. If the system does not reach the set-point for an unknown reason, the PI order tends to return maximum or minimum duty cycle values along the consecutive PI computation. In that case, the counters for maximum or minimum duty cycle reach their overflow value and then the converter may be stopped immediately.

This method ensures fast response from converter and shuts it down as for instance when a short circuit is present on V_{OUT} signal output. These counters are also used to detect that one operating mode has reached its own operating limits, and to accordingly manage the converter mode changes.

1.4.7 Getting started with the application

As soon as the buck-boost firmware has been programmed into the microcontroller, the application is ready to start. As mentioned in *Section 1.1: Required hardware*, an external power supply is needed for the demonstration and the mini-B USB cable connected on the PC side.

In order to start the converter, follow the steps:

1. connect the mini-B USB cable;
2. connect the external power supply to CN5 connector (V_{IN});
3. press the B2 button to reset the microcontroller.

The V_{OUT} target value is set by the firmware and can be modified manually for all other values included from 3 to 15 V_{DC}. This modification has to be performed in main.h file containing the V_{OUT} target parameter as shown below:

```
#define VOUT_TARGET ((uint16_t) 5000)
```

Please notice that this value is expressed in mV.

There is an optional update of the firmware that can be done, which consists in entering the real value of application 3V3 reference voltage. In the same main.h file, the following parameter (always expressed in mV) is mentioned:

```
#define REAL_3V3 (uint16_t) 3300
```

For accuracy purpose, this variable can be replaced by the current value of 3V3 voltage of the STM32F334 Discovery kit. A small variation on the 3V3 regulator can be detected, this parameter is used to account for it, and must be replaced by the actual value of the 3V3 that can be measured with a voltmeter connected in parallel with C19 capacitor. Otherwise, this value must be left as default.

There are 2 remaining constants in main.h file that can be precisely set for compensation of V_{IN} and V_{OUT} resistors bridges.

The user has to measure respectively V_{IN} and R45 voltages, and V_{OUT} and R46 voltages.

The VIN_RESISTOR_RATIO is obtained by computing (VR45/VIN) *10000, similarly the VOUT_RESISTOR_RATIO uses (VR46/VOUT)*10000.

For instance, the user sets precisely V_{IN} voltage to 10 V and reads the voltage value on R45, e.g. VR45 = 2.010 V. In this case VIN_RESISTOR_RATIO is (2.010/10)*10000 = 2010.

For V_{OUT}, the user reads both V_{OUT} and R46 voltage and computes similarly the VOUT_RESISTOR_VALUE.

New computed values are replaced in constants definition, otherwise, these values must be left as default ones as shown below:

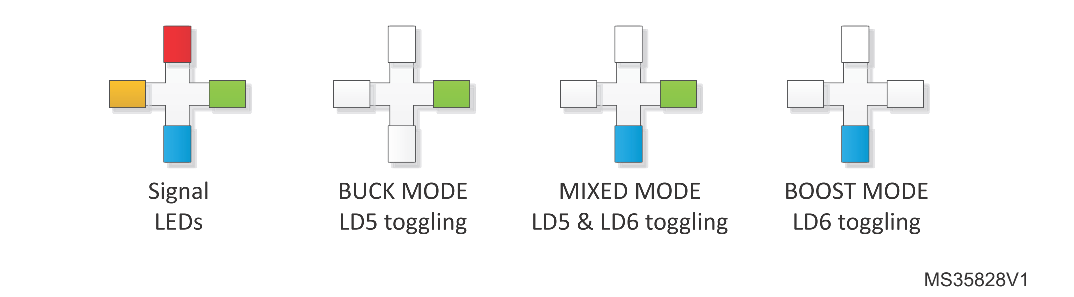
```
#define VIN_RESISTOR_RATIO(uint16_t) 2012

#define VOUT_RESISTOR_RATIO(uint16_t) 1988
```

A load or any electrical component complying with converter output characteristics can be connected to V_{OUT}.

Once the converter has started, the signal LEDs (green, blue, orange and red) are active and the following displays are observed according to V_{IN} / V_{OUT} conditions and to context:

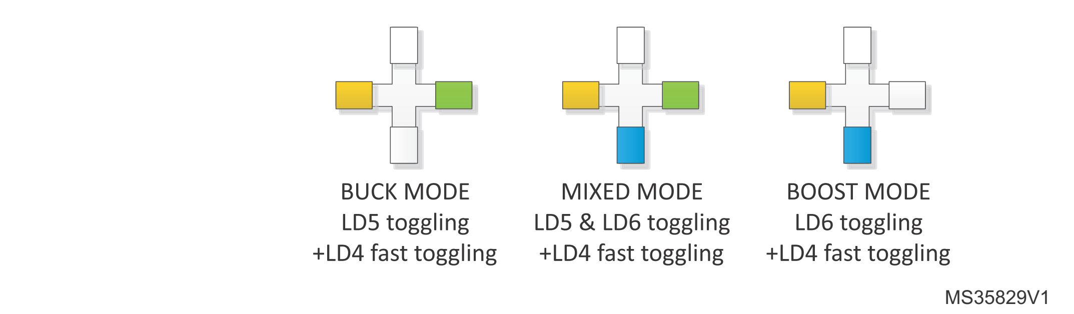
- Standard operation (*Figure 18*):
  - Green LED toggles for buck mode
  - Blue LED toggles for boost mode
  - both Green and Blue LEDs toggle for mixed mode.

Figure 18. Signal LEDs during standard operation



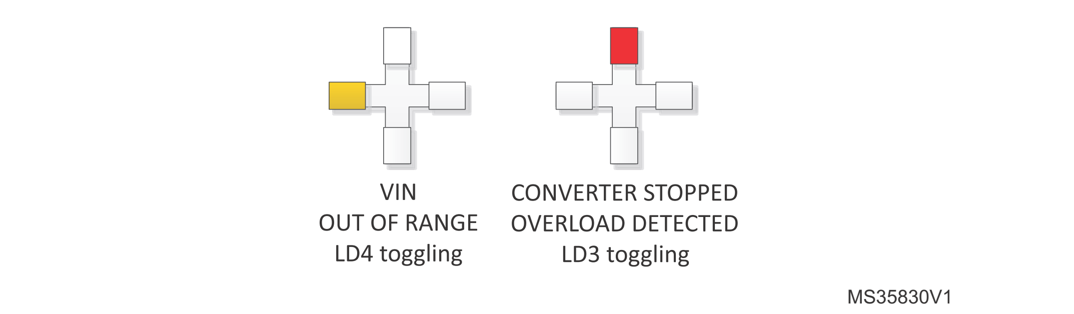
- Limited operation (*Figure 19*):
  - The same display is performed but the Orange LED is also toggling when converter operates near maximum duty cycles limits. Application can sustain such conditions (above 95% of allowed operation range), but user is warned that operating limits are close.

Figure 19. Signal LEDs during limited operation



- Stopped Operation (*Figure 20*): A fault condition has been detected into the converter:
  - Orange LED toggling:
    V_{IN} is not in the expected range [3;15] V_{DC}
  - Red LED toggling:
    Converter stopped (PI controller limits reached for a given time)
    Overload detected (current exceeds limit)
    Software initialization fault.

Figure 20. Signal LEDs during fault condition



2 Firmware description

2.1 STM32F334xx peripherals used by the application

This application example uses the following STM32F334xx peripherals with the settings described below:

GPIOs

GPIOs are essentially used for the 4 signal LEDS and few others for further development or debug purposes:

- PB6 to PB9 set as output GPIOs to drive the signal LEDs
- PA0 set as input GPIO and connected to the User push-button B1 but the button function is not implemented yet in this example. Users can place their own code into the mentioned section if needed.

ADC

ADC peripheral manages V_{IN} and V_{OUT} 12-bit ADC sampling. An ADC calibration is performed at application startup.

Two channels are used in injected conversion mode with different sampling times according to respective V_{IN} and V_{OUT} hardware adaptation. The ADC conversion is triggered by the HRTIM Timer A compare event CMP4 at every PWM period and the V_{IN} and V_{OUT} conversions stored.

HRTIM High-resolution timer

As mentioned earlier in overview section, the high-resolution timer is connected to the power interface of the DC/DC buck-boost converter.

The 4 HRTIM TA1, TA2, TB1, TB2 outputs are used for the 4 PWM signals generation of the H-bridge topology converter.

In debug mode, HRTIM Timer D is also used to highlight ADC conversion trigger event on TD1. “DEBUG” variable in firmware header must be set accordingly.

A HRTIM DLL calibration is performed at the beginning of HRTIM initialization function (duration 14 µs).

The HRTIM frequency is set to 250 kHz (period t = 4 µs) and determines the main switching frequency of the buck-boost converter.

The repetition counter for HRTIM Timer A is set to have an ISR every 8 PWM periods (4 µs x 8 = 32 µs). It can be easily lowered up to 4 if the overload protection is removed, this is freeing some MCU activity during interrupt routine.

The converter uses 4 different modes that need special initialization functions such as Idle, Buck, Mixed and Boost. By these dedicated functions, the HRTIM outputs are configured to select the set and reset sources combined with related events.

As a requirement of this application example, HRTIM Timer A and B use dead time configuration to set rising and falling dead times. This prevents overlap of switching command from MOSFET transistors from a same leg.

Please refer to *Section 1.4.4: Setting the STM32F334 High-resolution timer* for further information.

Interrupts

As a major process of this application firmware, the HRTIM Timer A interrupt HRTIM1_TIMA_IRQHandler() manages all relative tasks to the converter.

The PI functions are called from this interrupt routine and the converter duty cycles are updated accordingly. Depending on the ongoing operating mode, the converter also updates the ADC trigger at each interrupt. As soon as the converter reaches the limits of its operating mode, the mode change is performed during the interrupt.

A state machine is managing the LEDs display during the interrupt to highlight modes and other features.

If the overload detection is ON (#define OVERLOAD_ON is set), the duty cycle limit is computed into the interrupt every 1.6ms which correspond to 50 x 32 µs (repetition period of HRTIM1_TIMA interrupt). This task has been limited to avoid overloading the MCU at each repetition periods. As mentioned earlier in this document, the duty cycle limit is calculated with functions based on polynomial interpolation or regression lines. This is the major consuming task in terms of MCU resources even if this operation does not exceed 30% of MCU activity only every 1.6 ms.

2.1.1 Setting #define constants

There are 3 defined constants in header of the application firmware:

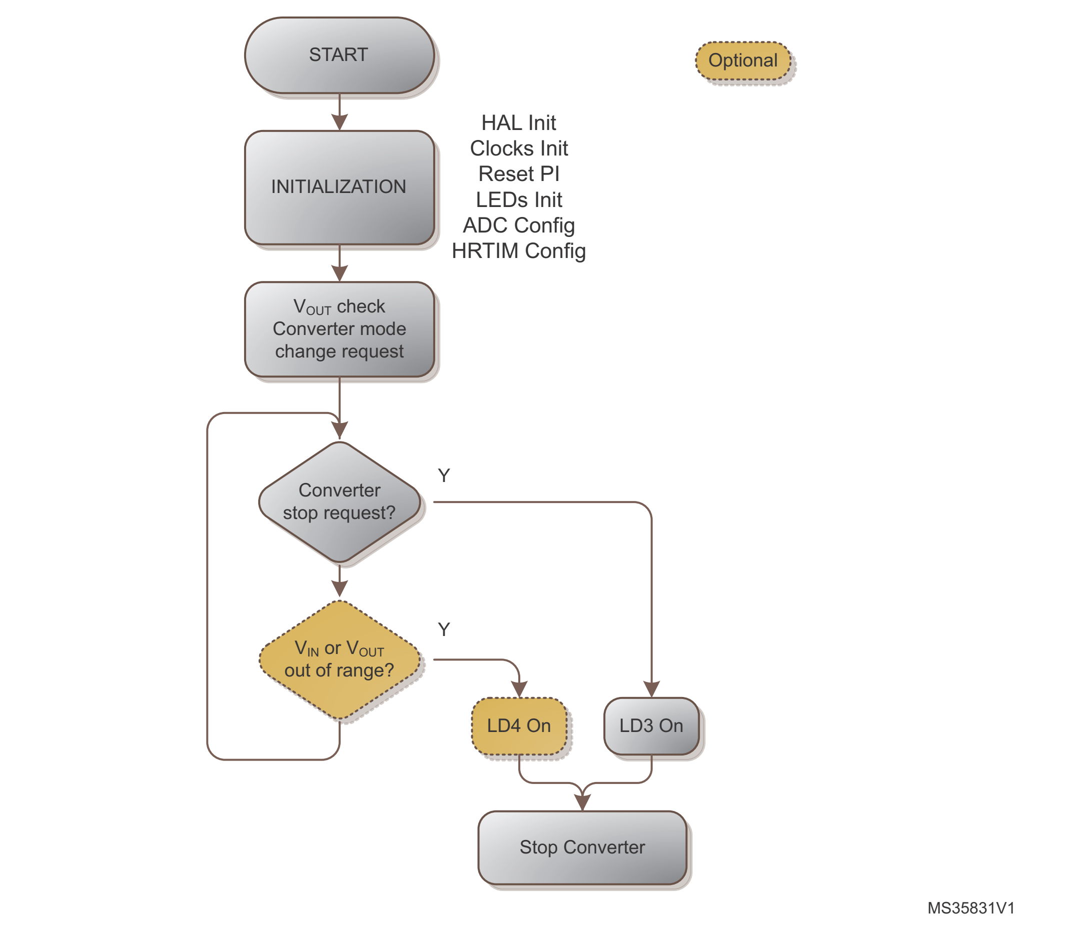
1. #define DEBUG: this constant is used to enable HRTIM Timer D and allowing the ADC trigger conversion event visible on TD1. This is only for debug purpose and to highlight interaction between timers' events.
2. #define RANGE_MONITORING_ON: this constant is used to enable the detection on V_{IN} and V_{OUT} voltages when one of these two voltages exceeds range limits (3 to 15 V_{DC}). If the application detects such eventuality for a given time, then the converter is stopped and the LED display is updated.
3. #define OVERLOAD_ON: this constant is used to enable the detection of any overload in the converter. The current duty cycle has reached the limit value for a given time (overload) and the converter must be stopped due to fault detection. The LED display is also updated.

2.2 Application flowcharts description

2.2.1 Application “main.c” flowchart

The “main.c” software (sketched in *Figure 21*) is limited to the initialization of the different peripherals used during this demonstration example. Mainly ADC and HRTIM peripherals are set at startup. Then the application firmware checks roughly what is the appropriate converter mode to start and puts a request mode change that will be serviced inside the interrupt routine. The application waits for any stop of the converter in case of fault detection. If any fault is detected during the HRTIM interrupt, then the converter stop request is set immediately but this stop is managed in the main sequence once the interrupt has been serviced and has ended. Different LEDs displays are performed.

Figure 21. “main.c” flowchart

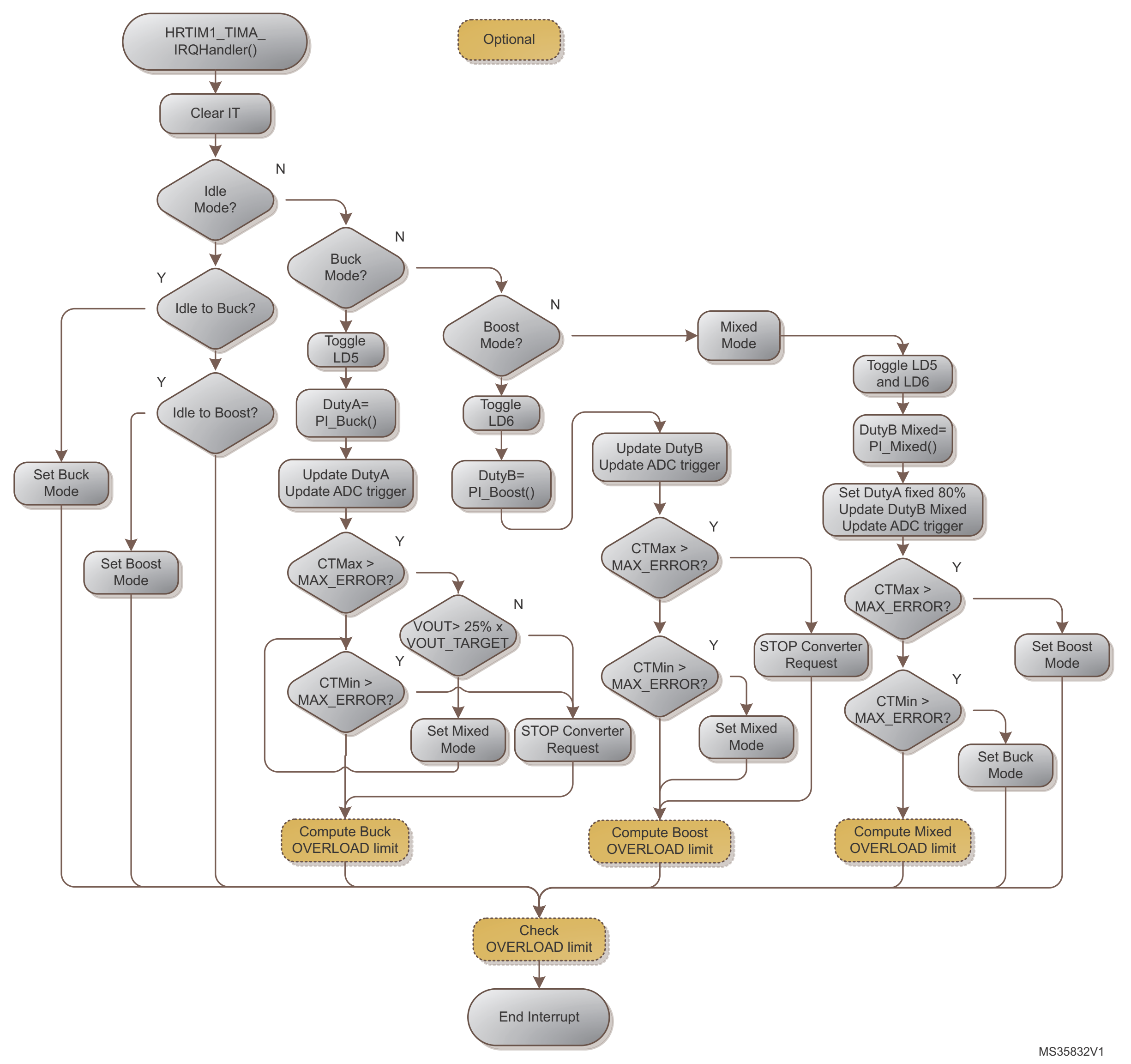


2.2.2 Application “HRTIM1_TIMA_IRQHandler()” flowchart

As shown in *Figure 22*, the major process is performed during HRTIM Timer A interrupt routine.

Once the interrupt has been serviced, the related IT is cleared immediately. The process identifies which is the running converter mode. Any change of the converter occurs in interrupt routine. This is to ensure smooth transitions between modes and applying the new waveforms in synchronization with the timer period. For each active mode such as buck, boost or mixed, a different display is managed and a dedicated PI function is called to compute the value of duty cycle. This duty cycle is updated at each interrupt and the ADC trigger event is set as well.

Figure 22. “HRTIM1_TIMA_IRQHandler()” flowchart



Then the converter checks for any information returned by the PI software controller that would overflow CTMax or CTMin counters. In these situations, a specific action is performed according to the current context. As for instance, in the buck mode converter, the application checks the CTMax counter overflow but also checks if the V_{OUT} output signal has started to rise at a minimum level. This is to distinguish a situation where the maximum duty cycle is requested to reach the target rapidly but also when the maximum duty cycle is applied and unfortunately V_{OUT} signal does not show any variation. In the first situation, the buck mode will change to the mixed mode while for the second one the converter will be stopped as a fault has appeared.

Basically, the duty cycle high and low limits for each mode are defined as main variables into the firmware based on the performances of this converter. As the 3 converter modes have adjacent operating area, the converter has to move from one mode to another when it reaches the operating limits of its own mode, precisely when counters CTMax and CTMin inform that a maximum or minimum duty cycle order is permanently returned by the PI controller. Any converter mode change is actually triggered by the overflow of CTMax or CTMin.

Regarding fault detection, the overload limit which is computed by the interrupt software is performed in two steps. The first step consists to collect the data from the relevant table and applying the right calculation according to the interpolation type. The second step is a common part of the interrupt task and checks for any duty cycle value that would exceed the overload limit. If the applied duty cycle operates near limits, then users are alerted by a display. If the applied duty cycle exceeds the overload limit for a specified time then the converter is stopped.

For further information on the firmware, refer to the complete firmware package available at http://www.st.com.

3 DC/DC converter main electrical characteristics

Table 6. Main electrical characteristics

Symbol	Parameter	Min	Typ	Max	Unit
V_{IN}	Input voltage (CN5)	3	-	15	V_{DC}
V_{OUT}	Output voltage (CN7)	3	-	15	
I_{in}	Input current on V_{IN}	0	-	0.55[1]	A
I_{out}	Output current on V_{OUT}	0	-	0.55[1]	
L	Inductor value	-	82	-	µH
I_L AVG	Average current in L inductor	-	-	0.75	A
I_{ripple}	Ripple current through inductor	-	0.3 x I_{out}	-	
V_{IN} min	Minimum Input voltage software detection	2.70	2.9	3.1	V_{DC}
V_{OUT} min	Minimum Output voltage software detection	2.70	2.9	3.1	
V_{IN} max	Maximum Input voltage software detection	14.9	15.1	15.3	
V_{OUT} max	Maximum Output voltage software detection	14.9	15.1	15.3	
V_{OUT} ripple	Output ripple voltage (I_{out} = 525 mA V_{IN} = 12 V_{DC}, V_{OUT} = 5 V_{DC})	-	100	-	mV_{p-p}
V_{OUT} accuracy	Output voltage accuracy (at 100% load)	-	0.5	1	%
f_{PWM}	Converter switching frequency	-	250	-	kHz
Duty cycle	PWM Duty cycle in buck mode	15	-	90	%
	PWM Duty cycle in boost mode	5	-	90	
	PWM Duty cycle in mixed mode	5	-	45	
Efficiency	I_{out} = 400 mA, V_{IN} = 12 V_{DC}, V_{OUT} = 5 V_{DC}	-	88	-	%
	I_{out} = 180 mA, V_{IN} = 7 V_{DC}, V_{OUT} = 12 V_{DC}	-	87	-	

[1] If overload detection is ON the maximum current is detected for this value +/- 10%.

4 Conclusions

DC-to-DC converters are widespread in industrial, consumers or automotive applications. They require high integration, efficiency, reliability but also more and more flexibility that analog solutions cannot always offer. Digital power conversions systems are taking the lead to address this market by offering a wide combination of technical solutions based on high-efficient digital timers.

This example around the STM32F334xx advanced ARM®-based 32-bit MCUs highlights some of the benefits of this microcontroller and especially the embedded high-resolution timer mainly designed for digital power conversions applications. It includes a very high range of settings that makes this buck-boost converter application example easier to develop. It also demonstrates its ability to setup and switch from one running configuration to another in one PWM cycle time.

This demonstration firmware combined with the low cost and high-performance STM32F334 Discovery kit are the easiest ever distributed tools on how to get started with STM32F334xx microcontroller and to experience its power-oriented solutions based on the brand new high-resolution timer.

5 Revision history

Table 7. Document revision history

Date	Revision	Changes
08-Sep-2014	1	Initial release.