STEVAL385LED4CH: four independent high brightness LED channels using the STLUX385A digital controller

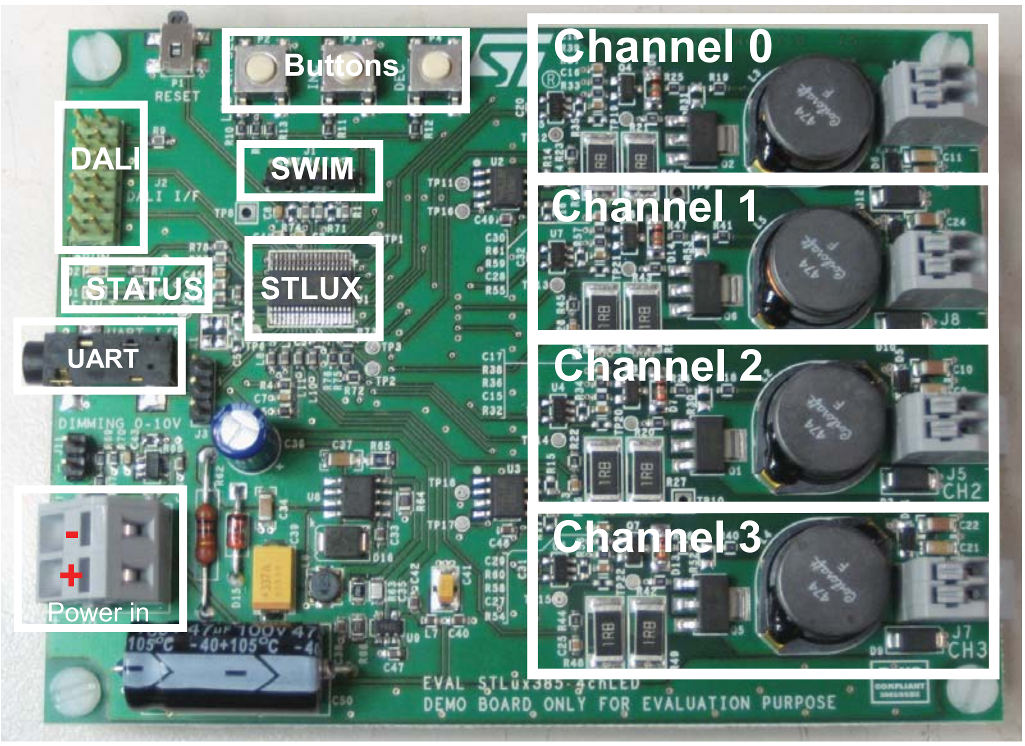
Introduction

The STEVAL385LED4CH demonstration board is a complete and configurable solution to manage four independent high brightness LED channels using the STLUX385A digital controller, which embeds advanced peripherals tailored to generate high resolution PWM signals. The LED current is independently regulated by the fixed-off-time (FOT) principle on each channel.

This document describes how to use the STEVAL385LED4CH board, the hardware and the firmware implemented on the board as well as the relevant measurements. The STLUX385A shipped with this board has already been programmed with firmware tailored for the STEVAL385LED4CH application and the source code is available.

The following picture gives an overview of the demonstration board diagram.

Figure 1. Demonstration board diagram



Contents

List of tables

List of figures

1 Demonstration board features

The STEVAL385LED4CH demonstration board implements the following features:

- Four independent LED channels (external LEDs)
- 3 to 10 LEDs per channel, connected in series
- Average LED current independently adjustable from 245 mA to 1065 mA with ~82 mA/step resolution
- LED brightness adjustable via PWM dimming (256 steps independent for each channel)
- Board power supply voltage (V_{LED_PW}) from 12 V to 48 V based on the longest LED string (external AC-DC power supply not included)
- Real-time fault detection and protection, such as open or short-circuit, independent for each channel
- Board configuration (chain current, LED numbers, …) accessible via USB-UART interfaces
- Three buttons to adjust the LED brightness for each channel
- Two status LEDs: green = ready-run, red = fault

Warning: The user is responsible for configuring correctly the board before connecting any LED string.

The user configures the right values according to the following parameters:

- Number of LEDs connected on each channel (from 3 to 10)
- Average LED current desired on each LED channel (from 245 mA to 1065 mA)
- 200 Hz PWM dimming duty cycle for each LED channel (from 0% to 100% in 256 steps)
- Board power supply voltage (from 12 V to 48 V)

The number of LEDs, the current and the PWM dimming working point can be set through the command interpreter accessible via the serial interface (*Section 7.3*). Alternatively, PWM dimming can also be manually adjusted via two on-board buttons.

The firmware installed on the STLUX385A automatically computes, for each LED string:

- Primary frequency upper and lower limits (400 kHz to 15 kHz with frequency step generated by PLL @ 96 MHz - auto-setup)
- Primary frequency duty cycle depending on the selected current, number of LEDs and power voltage

To guarantee the maximum reliability of the demonstration board against the most common LED failures, the STLUX385A implements the following software and hardware protections:

- For each LED string:
  - overcurrent
  - short-circuit
  - LED voltage drop falls below the 2.9 V limit
  - LED voltage drop higher than the 4.2 V limit
  - disconnected string (open circuit)
- Power voltage outside configured lower / upper limits

The STLUX385A device drives the 4 LED channels (CH0, CH1, CH2, CH3) seen on the right side of the board.

The board uses a DC-DC converter to supply the MOS drivers (U3, U2) with 10 V, while the STLUX385A (3.3 V or 5 V) is powered by a linear regulator.

There are three buttons to manually change the dimming period. The DALI and UART TTL interfaces are supported.

Two LEDs indicate the board status: the green one shows that the system is “ready” while the red one indicates a system fault or the intervention of a protection mechanism.

2 Getting started with the STEVAL385LED4CH demonstration board

This section aims to help designers to correctly power up and control the demonstration board in a short time. It describes how the board is connected to the load and how the serial command line and the board buttons control the light/color of the LED strings.

The STEVAL385LED4CH board provides 4 independent LED channels and the user can connect up to 4 different LED chains (not included in the STEVAL385LED4CH package). The board is configured to control standard LEDs with a forward voltage (VF) between 2.9 ("`V_LMIN`") and 4.2 ("`V_LMAX`") on each LED. The limits are defined in the firmware and they can be changed by modifying the source code (see [`#define V_LMIN`] and [`#define V_LMAX`] into "led.h"). Alternatively, the user can simulate the required voltage drop by configuring, via the (ln command) an apparent incorrect number of LEDs.

The LED numbers and the current associated are configured via the serial command defined in *Section 7.3*.

The board accepts a DC input power in the range 12 V - 48 V. The user chooses the correct input voltage depending on the LED numbers and the voltage drop. The firmware accepts a minimum voltage of 2.8 V ("`V_COM_MIN`") across the inductances when the MOSFET is ON. If an incorrect power voltage value is set up, the LED current is not regulated correctly but the STLUX385A prevents the external LED module from damage. The only way to damage an external LED module is to configure a current over the maximum accepted by the LED module or define a number of LEDs different from those used. When a new LED module is applied to the board, it is recommended the user to start with the minimum current ("lc x 0") and with the dimming OFF ("ll x 0). Note that the firmware stores the last serial commands (LED number, LED current, LED dimming and adaptive voltage compensation enable/disable) on the internal E^2PROM and applies those values at power-on or reset.

2.1 First power-on procedure

The following section describes a step-by-step procedure, which guarantees a correct power-on during the first configuration of the board.

The guide assumes that the following elements are available:

- STEVAL358LED4CH board
- USB TTL serial cable shipped with the board
- Power supply
- Computer running Windows® operating system
- LED string

It is recommended the LED string to be disconnected before the first startup.

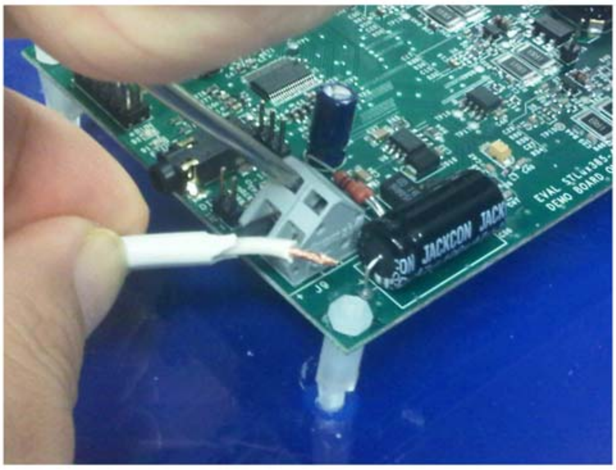
The first power-on instructions as follows:

1. Connect the USB TTL serial cable to the USB port of the computer. Windows recognizes the cable as a TTL and launches the driver installation.
2. Install the appropriate USB TTL serial cable drivers.The drivers are the "virtual COM port (VCP)" and they are provided by FTDI (http://www.ftdichip.com). Should you need any help install the drivers, please refer to your IT administrator. Once the drivers are

installed, the Windows device manager panel shows a new USB serial port (COMx) device and the COM number associated.

3. Verify that the power supply output voltage is within the board specification.
4. Switch OFF the power supply.
5. Connect the power supply to the board and check the input polarity. The following picture illustrates how to connect the power cables:

Figure 2. Power supply connection



AM16786v1

6. Power on the board and check that the power supply current level is within the appropriate levels.
7. Monitor the red status LED on the STEVAL358LED4CH demonstration board. During the startup, the red status LED goes ON for few seconds before switching OFF. Should the red status LED stay ON for more than 3 seconds, the STLUX385A device is indicating a problem. The status LED should be checked after every step of the power-up procedure. The STLUX385A indicates any issue encountered: short-circuit, overcurrent, input power under the limit, etc. Note that, once the status LED is red, it can only be switched OFF via the "co" console command.
8. Connect the USB TTL serial cable to the STEVAL358LED4CH board.
9. Run Hyperterminal or an equivalent program from the computer connected to the USB TTL serial cable. The connection configuration is as follows:
   – baud rate 115200
   – data bits: 8
   – stop bits: 1
   – parity: no
   – flow control: no handshake

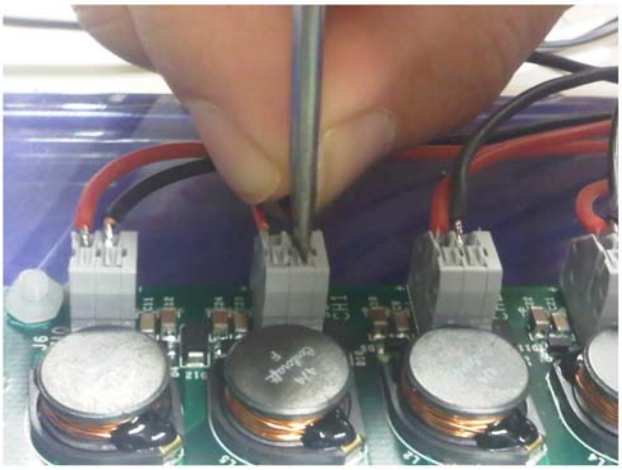
In order to verify that the connection is valid, type "?" and press ENTER. If the link is ok, the STLUX385A list of commands is displayed.

10. Should the red status LED list a problem, use the "st" command and look at *Table 3* to find the root of the problem.
11. Configure the LED string parameter via the "lc" (LED current) and "ln" (LED number) commands. See *Section 7.3* for the command reference. Note that any new setting is stored in Flash and is automatically used during the next startup.

Warning: The current level has to be configured according to the LED capability. A current level higher than the one supported by the LED may damage the LED itself.

12. Switch OFF the dimming using the "ll x 0" command where "x" is the channel number selected to drive the LED string. Once the dimming is disabled, channel x doesn't generate current.
13. Connect the LED string to the appropriate channel "x" connector. Please observe the LED polarity. The following picture illustrates how to connect a LED string.

Figure 3. LED string connection



AM16787v1

14. Increase the dimming level using the "ll" command and observe the LED string switching ON.
15. Enable the adaptive voltage compensation using the "au x 1" command.

3 Design concept

The STEVAL385LED4CH demonstration board is based on a modified buck topology (reversed buck) and the fixed-off-time (FOT) principle is used to regulate the current into each LED string. The advantage of the FOT principle is that the high frequency current ripple and, therefore, the average LED current depend only on the off-time and the LED output voltage. With a fixed-off-time, the output voltage depends on the temperature and the number of LEDs. To compensate this effect, the STLUX385A implements the adaptive voltage compensation algorithm, which constantly monitors the LED voltage drop and adjusts the off-time. The input voltage variations do not affect the average current.The average LED current is directly controlled by regulating the LED peak current (configurable), which is sensed during the MOSFET on-time.

While the off-time parameter is fixed, the on-time is a function of the time that the current takes to reach the selected level: this time depends on the LED current, power voltage and LED voltage. The firmware implements the following two functions to compute the OFF and ON-time:

Equation 1

$$T_{OFF} = \frac{(2 \times (I_{MAX} - I_{AVR})) \times L}{V_{LED}}$$

Equation 2

$$T_{OFF} = \frac{(2 \times (I_{MAX} - I_{AVR})) \times L}{V_{LED\_PW} - V_{LED}}$$

where:

- V_{LED}: output voltage of the LED string (equal to V_{LED_PW} -V_{COMx} signals on the schematics)
- I_{MAX}: maximum (peak) current imposed by user and defined by *Table 1* (near 92 mA/step)
- I_{AVR}: current imposed by design (I_{MAX} - 10%)
- L: inductance imposed by design (470 µH)
- V_{LED_PW}: input voltage present at the main power pin

The on-time calculated in equation 2 has to be considered as an ideal value: the real one is actually a consequence of the triggering of a comparator monitoring the peak value of the LED current. For this reason, a percentage of the calculated on-time is added (T_x). This is empirically imposed by adding 20% to the on-time and doubling the resulting value.

Equation 3

$$T_{ON(MAX)} = (T_{ON} + T_X) \times 2$$

Equation 4

$$T_{ON(MAX)} = (T_{ON} + (T_{ON} \times 0.2))x2 = 2.4xT_{ON}$$

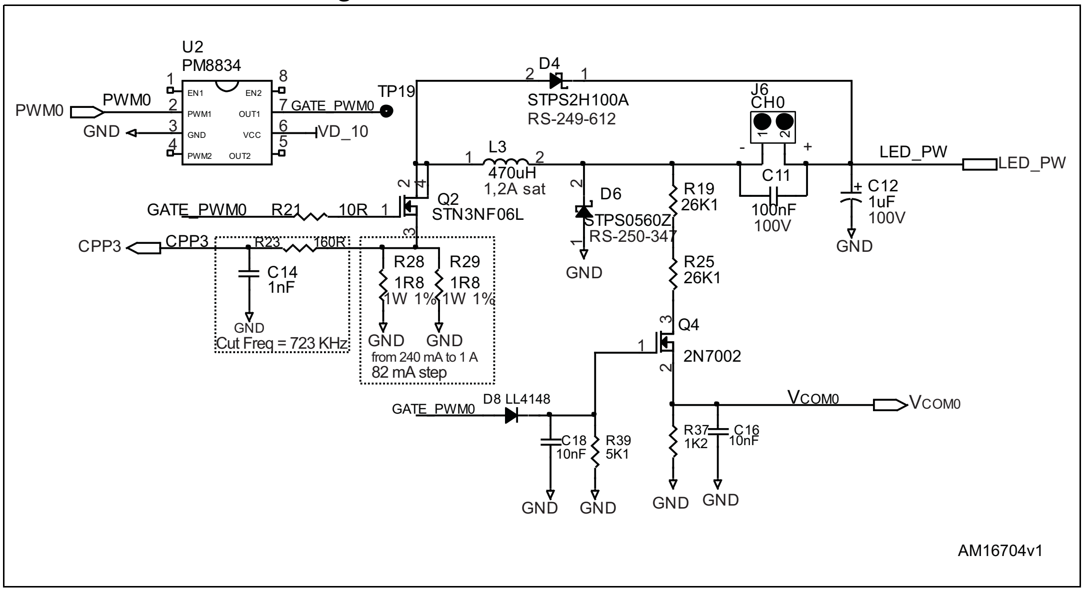
Where T_x is the time to search current and it is equal to 20% of T_{ON}. The extra time has two effects: the former, during the dimming operation, allows a quicker response from OFF to ON transition; the latter prevents a frequency lower than 15 kHz (in the audible spectrum). On the board the circuit for the transition mode principle is also present but it is not described in this manual.

3.1 LED channel schematic

To implement the FOT principle and to use the STLUX385A microcontroller, the following schematic (single LED chain) is used. It represents a standard reverse buck structure. The STLUX385A applies the PWM frequency via the P_{WM0} signal and monitors the voltage of the V_{COM0} signal. It also monitors the LED current during the "on-time" by sensing the voltage present at CPP3 line. The V_{LED} signal is calculated by the subtraction from V_{LED_PW} and V_{COM0}. The V_{COM0} signal is acquired 100 µs after the PWM dimming starts.

The schematic for a single LED channel is reported in *Figure 4*:

Figure 4. Schematic - LED channel 0



Explanation:

- The LED chain is connected to the J6 connector.
- The MOSFET Q2 drives the current from LED_PW to the LED channel and the inductor L3, charging it. The Inductor L3 is defined for all selectable current ranges. Should the user change the inductor values, some parameters, described in this manual, such as the K constant, have to be recalculated.
- The resistors R28 and R29 act as shunt resistor to sense the current through the LEDs. The current signal is then fed to CPP3 pin through an RC filter represented by the resistor R23 and the capacitor C14. The filter removes the spike caused by the turn-on current generated by MOSFET Q2. The STLUX385A detects the peak current by comparing CPP3 with an internal, DAC generated, voltage.
- The R19, R25, Q4 and R37 are used to feed the voltage of the LED chain cathode to the STLUX385A's ADC. The MOSFET Q4 is used to prevent the small biasing current

of the voltage divider from flowing through the LED during the dimming off-time. The ADC maximum voltage is 1.25 V (the ADC gain is configured to x1).

- The D4 diode is active during the OFF period, when the voltage across the inductor goes high.
- The U2 circuit (PM8834 MOSFET driver) is used to generate the correct gate voltage in case the STLUX385A is powered from 3.3 V.
- The D6 diode is used to protect the circuit if the LED chain opens or is accidentally removed during the normal operation.

3.2 Solving the equations

The T_{OFF} value introduced in the previous equation has to be converted into a number that could be used by the STLUX385A. In particular, T_{OFF} is implemented in the SMED as a timer. This timer is a 16-bit register and is clocked according to the SMED clock (CLK = 96 Mhz). The timer register value is equal to:

Equation 5

$$T_{OFF(SMED)} = T_{OFF} \times CLK$$

where:

- $T_{OFF(SMED)}$: timer value
- CLK: SMED operating frequency

The STLUX385A monitors the voltage drop caused by the LED and it is calculated as the difference between the power voltage (V_{LED_PW}) and the voltage measured after the LED string (V_{COMx}). Both V_{LED_PW} and V_{COMx} are read by the ADC unit.

Equation 6

$$V_{LED} = \frac{ADC_{LED\_VAL} \times ADC_{FS} \times P}{ADC_{TOP}}$$

where:

ADC_{LED_VAL}: the sensed LED drop voltage and calculated as V_{LED_PW} - V_{COMx}

V_{COMx}: ADC value read from middle point of each channel (LED-inductor)

V_{LED_PW}: ADC value read from the main power

P: is the resistor partition gain $(R_{19}+R_{25}+R_{37})/R_{37}$ = (53400)/1200= 44.5 Ω

ADC_{FS}: ADC top voltage = 1.25 V

ADC_{TOP}: is the step value of ADC (2^{10}) = 1024

Equation 1 can be rewritten as follows:

Equation 7

$$T_{OFF(SMED)} = 2 \times (I_{MAX} - I_{AVR}) \times L \times \left(\frac{ADC_{TOP}}{ADC_{LED\_VAL} \times ADC_{FS} \times P}\right) \times CLK$$

for example:

Equation 8

$$T_{OFF(SMED)} = \frac{2 \times (I_{MAX} - I_{AVR}) \times L \times ADC_{TOP} \times CLK}{ADC_{LED\_VAL} \times ADC_{FS} \times P}$$

where:

- I_{MAX}: max.current imposed through the DAC setup (1/16)
- I_{AVG}: average current - imposed as 90% of I_{MAX} (by design)
- (I_{MAX} - I_{AVG}): current difference: (I_{MAX}*0.1)
- I: is the inductance value = 470 10^{-6} H (by design)
- CLK: Is the SMED CLK = 96 10^{6} Hz

The average current I_{AVG} is set by design to be equal to 90% of I_{MAX}.The algorithm controls the current to be centered in I_{AVG}, with a ripple equal to +/- 10% of I_{MAX}. In equation 8, the current and the voltage across LED are unknown. The current (I_{MAX}) is an input value defined by the user. The voltage on the LED string is the difference between the main power voltage and the V_{COMx} voltage sensed by two ADC channels. Considering 1.184 A I_{MAX} current and I_{AVG} approximated to 1.066 (the real value is 1.0654), the equation 4 is expanded into:

Equation 9

$$T_{OFF(CLOCK)} = \frac{2 \times (1.184 - 1.066) \times 470 \times 10^{-6} \times 1024 \times 96x10^{6}}{ADC_{LED\_VAL} \times 1.25 \times 44.5} =$$

$$= \frac{196763}{ADC_{LED\_VAL}} = \frac{K}{ADC_{LED\_VAL}}$$

where K (196763 in this case) is a constant for the given LED current selected by the user. The I_{MAX} values accepted by the STEVAL385LED4CH board are bounded by the DAC limits as shown in the following table:

Table 1. Current, DAC, K relation table

Index	DAC value	Medium current(1)	Maximum current	K (dec)
10	13 (0x0D)	1065 mA	1184 mA	196.763E+03
9	12 (0x0C)	984 mA	1093 mA	181.628E+03
8	11 (0x0B)	901 mA	1002 mA	166.492E+03
7	10 (0x0A)	819 mA	911 mA	151.356E+03
6	9 (0x09)	738 mA	820 mA	136.221E+03
5	8 (0x08)	648 mA	729 mA	121.085E+03
4	7 (0x07)	574 mA	638 mA	105.949E+03
3	6 (0x06)	492 mA	547 mA	90.814E+03
2	5 (0x05)	410 mA	456 mA	75.678E+03

Table 1. Current, DAC, K relation table (continued)

Index	DAC value	Medium current(1)	Maximum current	K (dec)
1	4 (0x04)	329 mA	364 mA	60.543E+03
0	3 (0x03)	245 mA	273 mA	45.407E+03

1. Medium current: the value has been approximated to the nearest unit.

The index level is used by command "Ic"(see *Section 7.4.1*) to select a current value.

The output of equation 9 is the value to be stored in the SMED register that represents T_{OFF}.

Since K is constant, as long as the user doesn't change the current index level, the STLUX385A updates the T_{OFF} value only when the ADC_{LED_VAL} changes, when the LED voltage drop changes. The only operation required to update T_{OFF} is a 32-bit division. Equation 4 can be developed as follows:

Equation 10

$$T_{ON(MAX)} = T_{ON} \times 2.4 = \frac{2 \times (I_{MAX} - I_{AVG}) \times L \times ADC_{TOP} \times CLK}{ADC_{VCOM\_VAL} \times ADC_{FS} \times P} \times 2.4 =$$

$$= \frac{2.4 \times K}{ADC_{VCOM\_VAL}}$$

Since "K" is constant during operations, $T_{ON(MAX)}$ can be updated with a single 32-bit division. SMED implements the $T_{ON(MAX)}$ timer as two consecutive timers operating in different states (S1, S2).The overall update of T_{ON} and T_{MAX} timers requires 3 divisions plus the time to update the relevant SMED shadow registers.The STLUX385A needs just 200 µs to complete the operation.The "K" parameters are calculated to satisfy a board operating with a 12 V - 48 V input power range and support from 3 to 10 LEDs. Should the user use either different inductances or different parameters such as the ΔI current, only the "K" constant has to be recalculated and the FW has to be updated with the new K indexes.

4 SMED implementation

This section describes how the FOT (fixed off-time) principle is implemented in the SMED (state machine event driven). Each SMED generates the correct PWM signals and the SMED state evolution depends on both external events and internal timers.

Note: Please refer to the STLUX385A product documentation for a detailed explanation of the SMED technology.

The SMED is responsible for the generation of the PWM necessary to maintain the I_{AVG} current through the LEDs. When the SMED is triggering the PWM output, the LEDs are ON.

Digital dimming is achieved by applying a duty cycle to the LEDs. The software controls the duty cycle (referred as dimming in this document) by suspending and restoring the SMED's PWM generation. When the SMED stops triggering the PWM line, the LEDs are OFF.

Note that the dimming frequency is much lower (~200 Hz) than the PWM frequency used for current control (70 Khz - 400 kHz).

4.1 SMED input matrix

Each SMED has three associated signals. Only two inputs are used in the FOT principle described in this manual. The inputs have been logically mapped to represent:

- SMED_in(0) - digital input: senses the inductor null current (currently not used).
- SMED_in(1) - comparator input: detects when the current has reached the DAC value.
- SMED_in(2) - software input: starts and stops the SMED. Used for dimming control.

Table 2 summarizes the input and event relationship:

Table 2. SMED input association

	LED CH	SMED_in(0)	SMED_in(1)	SMED_in(2)
SMED 0	0	DIGIN 2	DAC-CPP 3	SW 0
SMED 1	1	DIGIN 1	DAC-CPP 2	SW 1
SMED 2	2	DIGIN 4	DAC-CPP 0	SW 2
SMED 3	3	DIGIN 3	DAC-CPP 1	SW 3
SMED 4	Not used	DIGIN 0		SW 4
SMED 5	Not used			SW 5

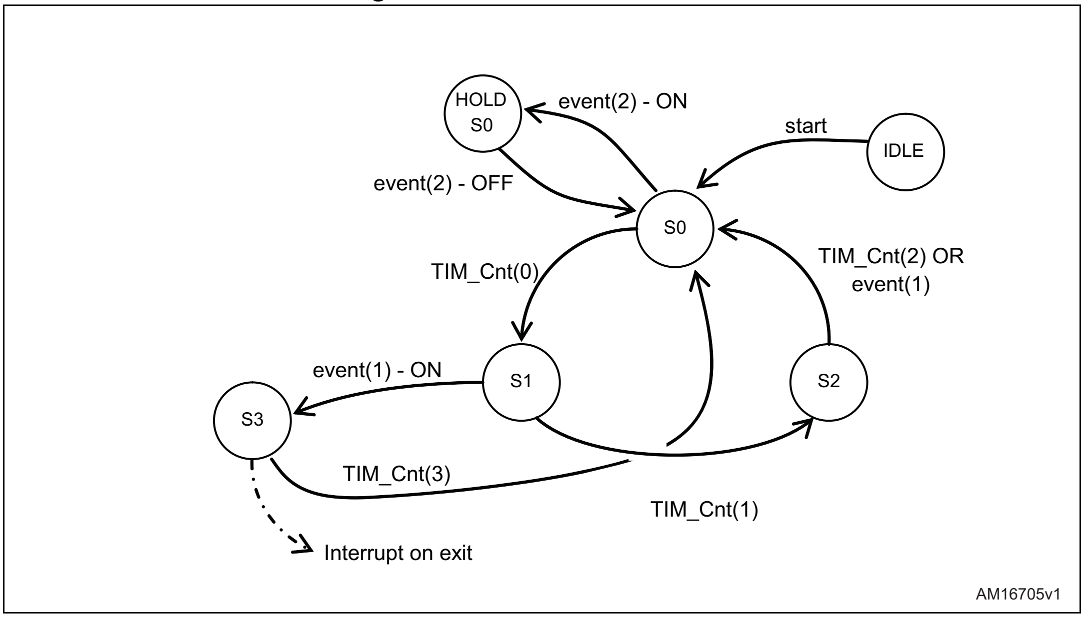
Option:

- SMED 4 PWM output is optional.
- SMED 5 PWM output is filtered and connected to CPM3. SMED5 may be optionally used to generate a reference signal for the CPM(3) input. CPM3 can be used to sense and detect the top-current event for LED channel 0 and increase the string average current granularity.The possible current steps are equal to 5 mA (typical). This option is not used in the current firmware version.

4.2 State description

The STEVAL385LED4CH demonstration board uses four independent SMEDs, one for each LED string. All the SMEDs use the same configuration and share the same control algorithm. This section describes a single SMED but the same principles are applied to all the SMEDs. The following diagram shows the SMED state evolution:

Figure 5. SMED state evolution



A state is defined by the Sx symbol while a transaction between two states is represented by a line. A transaction is generated by an event, displayed as text on the line. The event TIM_Cnt (x) is generated by the timer when it reaches the value configured in State x. An event(line) represents an event generated by the input line. The inputs associated to a SMED are defined in *Section 4.1*.

Each SMED has 3 associated events, marked as event(line) in *Figure 5*. Each event can be: type Event_Seq or Event_Jmp. While Event_Seq events push the state machine to the next sequential state, Event_Jmp allows the state machine to reach any state.

Each state is here described:

- IDLE: initial state

  This is the SMED initial state. The PWM line is set to 0 to keep the MOSFET OFF. The next state is S(0) and it is automatically entered when the SMED starts. On exit: clear timer AND set PWM to ON.

- S(0): off-time

  This state implements the FOT time and the output PWM line is set at 0 V. S(0) enters the HOLD state when the SWx bit (event 2) is set to 0 or it goes to S(1) if the timer reaches the FOT time. The SMED timer is cleared on exit.

  Event_Jmp = Hold jump ON = event(2).

  Event_Seq = TIM_Cnt.

  On exit: clear timer AND set PWM to ON.

- S(1): fault zone

  This state is used to prevent damages due to an external malfunction (either a hardware problem or an extreme fluctuation of the input voltage). The PWM output line is set high during the transaction between S(0) and S(1) and turns the MOSFET ON, leaving the current flow through the LEDs and the sense resistor. In normal conditions, the current shouldn't reach the peak during S(1) and the S(2) timer overflows, pushing the SMED to S(2). The S(1) time is set to be equal 33% of the total T_{ON} state (S(1) + S(2)) maximum time. In exceptional conditions, the CPPx signal arrives earlier, the S(1) timer expires and the SMED evolves to S(3) (OCP). CPPx indicates the overcurrent in the circuit. The SMED timer is cleared on exit.

  Event_Jmp = event(1).

  Event_Seq = TIM_Cnt.

  On exit to S(3): clear SMED timer; set PWM to OFF.

  On exit to S(2): do not clear SMED timer; turn ON the PWM output.

- S(2): current limit

  This state is used to find the current peak in normal conditions. The current peak is imposed by the voltage configured on the DAC. When the CPPx event is detected, the SMED advances to the S(0) state. During this state, the output PWM line is high and the MOSFET is ON. A timer overflow indicates that the current has not reached the selected peak, for example because the LED string is not attached. This time is computed by the (66% + 20%)*2 equation 3. On exit, the PWM output is set to 0 V, switching OFF the MOSFET. The next state is the non-sequential state S(0).

  Event_Jmp = N/A.

  Event_Seq = Event(1) OR TIM_Cnt ("and/or" bit guarantees a non-sequential jump).

  On exit to S(0): reset timer, set PWM to 0.

- S(3): overcurrent protection

  When Event_Jmp is received during state S(1), the SMED has found an OCP (overcurrent protection) event and goes to S(3). During the transaction, the PWM output line is set to 0 V in order to turn the MOSFET OFF. The S(3) timer is set to its maximum value (0x1F0 = 5 µs) to protect the hardware. When the SMED timer overflows, the SMED advances to sequential S(0) state. The transaction also generates an internal interrupt and the software can monitor the OCP and stop the SMED evolution.

Event_Jmp = N/A.

Event_Seq = TIM_Cnt.

On exit to S(0): reset timer, generate SW interrupt.

- HOLD S(0)

  This state starts from S(1) when the software needs to stop the SMED evolution, due to the dimming duty cycle OFF time. HOLD is also used when OCP is detected during S(1). The PWM output line on HOLD state is 0 and the MOSFET is OFF. When the FW sets the SWx bit to 1, the SMED goes back to S(1).

Note: The register associated to the SMED is updated when either the SMED is on HOLD state (during the dimming off-time) or the output PWM line is zero (S0 state).This approach guarantees that the PWM output is always set to 0 when the new parameters are applied.

4.3 Input/output signals

The SMED uses the following list of input/output signals and pins:

- PWMx: PWM output signal

  The signal is fed into the MOSFET's driver input to control the MOSFET's switching state. The PWM operates according to the T_{ON} and T_{OFF} time faced in *Section 2*. When the PWM is high, the MOSFET is ON and vice versa. When the supply voltage of the STLUX385A is 5 V, the output signal can be directly connected to the MOSFET. Instead, when the supply voltage is 3,3 V, the PM8834 buffer is required.

- CPPx: comparator input signal

  CPPx receives the current sense signal and compares it to the internal DAC value. When the voltage on CPPx pin is higher than the DAC value, the output of the internal analog comparator goes high, and vice versa. The comparator output is the event(1) internal signal. Event(1) is programmed at level mode so that the SMED can detect immediately the eventual anomaly during S(1). If edge mode was used and the anomaly happened outside S(1) then S(1) wouldn't receive the edge event. There are only two SMED states that check for this event: fault zone (S1) and current limit (S2).

- SWx: software event

  Used to start and pause the SMED. When the SMED is paused, the PWM output has to be set to 0. The SWx event is used during S(0) to pause the SMED and during HOLD to restart. The event is programmed at level mode and is used by the software to define the dimming duty cycle period. Thanks to the event mode, the SW trigger is detected by S(0) even if the SW edge happens while the SMED is in a different state then S(0). In this case, the SMED completes the cycle, reaches S(0), captures the SW level and goes on HOLD.

- DIGINx: zero current detection

  The pin is used for zero current detection while it is in transition mode. At the moment the transition mode is a proposal and the pin is not managed.

- V_{COM0}

  The voltage applied to the pin is read by the STLUX385A ADC and it is used to compute the T_{ON} time and the T_{OFF} time. The voltage is captured 100 ns after the beginning of the ON portion of the dimming cycle as the current is expected to reach the correct value.

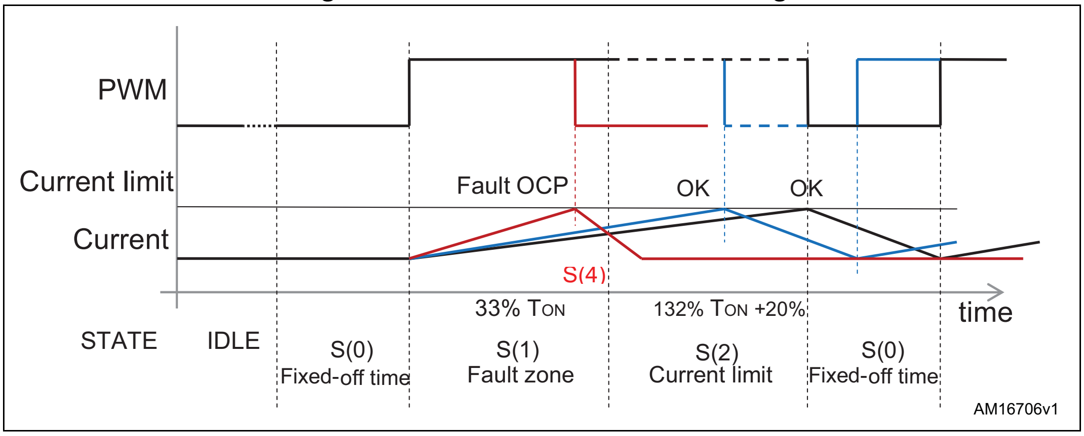
- V_{LED_PW}

  The voltage applied to this pin is read by the STLUX385A ADC and is used to compute the T_{ON} time and the T_{OFF} time. It is acquired at the same time of the V_{COM0} voltage using the 8 values, four for V_{COM0} and four for V_{LED_PW} - interleaved, stored by the ADC in registers ADC_DATH< 7-0 > and ADC_DATL< 7-0>.

4.4 SMED state time evolution

The following diagram describes the SMED state time evolution. Only one PWM period is illustrated.

Figure 6. SMED state evolution. Timing



The colors identify the case of a correct or incorrect situation:

- The black trace is the output current observed during a correct setup where the SMED evolution is triggered by timer events only. In this case, no event_Seq or event_Jmp are received; note that S2 may not have reached the peak level during S(2).
- The blue trace represents the situation where the current reaches the limit point (event_Seq) during S(2).
- The red line shows the SMED evolution when an OCP event (event_Jmp) is detected during S1. The SMED is programmed to generate an interrupt when an OCV condition happens. The main program receives the interrupt and stops the SMED.

5 Firmware implementation

The following section offers a view of the firmware implementation. For more details about the firmware, refer to the source code. The compiled portion of the firmware core is 3.5 Kbyte and the code size is ~12 Kbyte. Most of the code size is allocated to the debugging machine and general user interface (see *Section 7.3*).

The PWM algorithm and the OCP controls are entirely implemented in the SMED block, leaving the CPU free for control tasks such as: slow varying algorithmic compensations, the command line, and interrupt management.

5.1 Dimming algorithm

The dimming algorithm controls the brightness of the LED string by modulating the amount of time, the LED is switched ON during a ~5 ms period. This period defines the dimming cycle, which operates with a ~200 Hz frequency. The longer the LED is ON during a single cycle, the brighter the LED results to human eyes. To turn ON a LED, the algorithm simply activates the SMED associated by setting the SMED's SWx event to active state.

Once the desired on-time period is passed, the LED is turned OFF and the algorithm waits for the beginning of the next dimming cycle to turn ON the LED again.

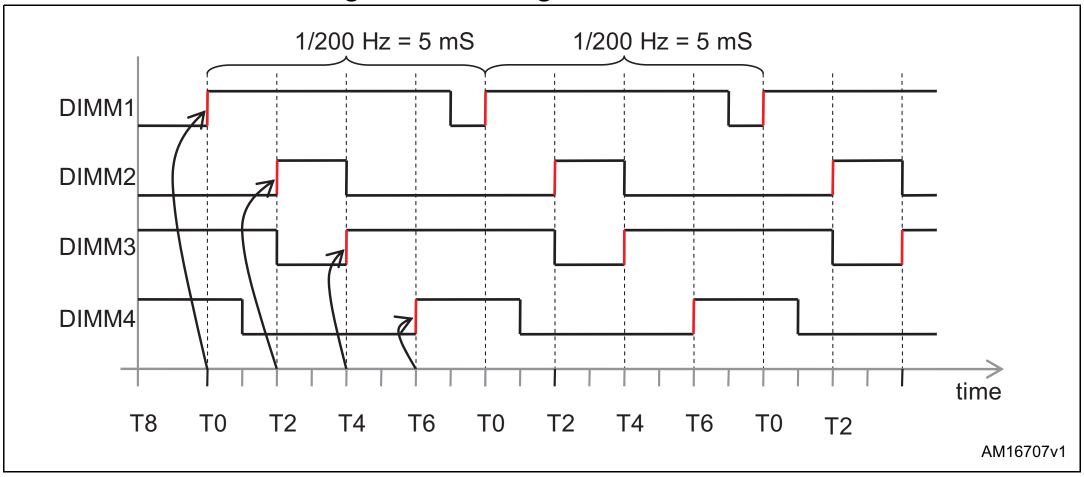
To optimize power, the dimming cycle in channel x is 90 degrees shifted respect to channel x-1. LED chains can be configured with different on-time length; however the dimming start period is fixed by design. In this way, the LEDs are always turned ON at different times, reducing current peaks on the power supply.

The LED on-time is instead a multiple of 20 µs and is based on the STMR timer, which itself has 1 µs granularity.

The software stores the time-on length in the `onoff[8][2]` structure, where the first byte indicates the channel and the PWM action (turn ON or OFF the LED) and the second byte specifies the time (up to 256 time units = 256 x 20 µs = 5120 µs) before the next event.

The next example shows the activity of LED channels with different dimming on-times: DIMM1 is active at 87,5%, DIMM2 is active at 25%, DIMM3 is active at 75% and DIMM4 is active at 37.5%. When the dimming is ON the SMED output PWM is active and the LED light is ON.

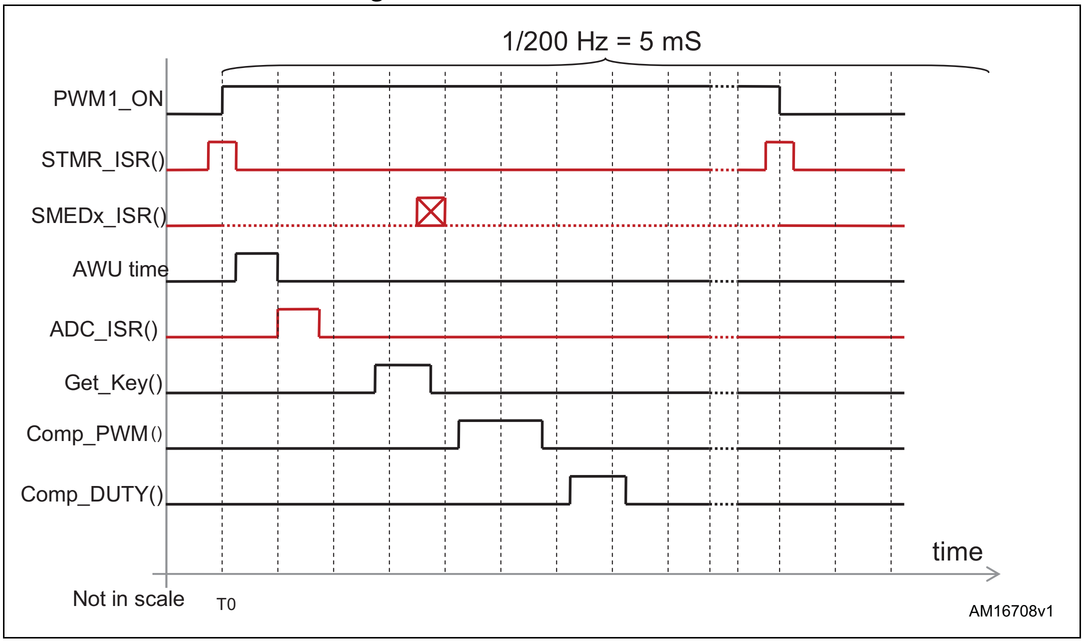
Figure 7. Dimming channel relation



5.2 Firmware evolution

The following section describes the software timing characteristics as shown in the following picture. The picture shows the relationship between two consecutive `STMR_ISR()` routines and the other principal subroutines. The colors define if the routines run in interrupt mode (colored) or are called from main (black). The PMW1_ON line is the dimming period: when the line is high, the PWM signal is high too.

Figure 8. FW timeline evolution



The lines are here described:

- `STMR_ISR()`: STMR timer interrupt.

  The STMR timer has a granularity of 20 µs (STMR unit). The timer is used to start and stop the dimming period for each SMED and triggers the SWx bit on the MSC SMSWEV register. On every interrupt, the timer is reprogrammed so that the next interrupt is received when the next dimming even is required, as specified in the `onoff[][]` structure.

When the interrupt routine sets a SMED ON, the startup flag for the ADC delayed acquisition (using the AWU peripheral) is also set.

- `SMEDx_ISR()`: SMED interrupt

  The interrupt is used to stop the SMED when an OCP problem is detected. The interrupt is generated by a SMED on exit from S3. The SMED is in S3 only if event(1) is received during S1. The interrupt routines stop the SMED by changing the SWx bit and giving the fault signal to the main function, which turns on the red "fault" LED. The SMED is reactivated on its next dimming cycle and if the OCP problem is not removed then the SMED evolves into S3, triggering the `SMEDx_ISR` interrupt again.

  Auto-wakeup unit. The auto-wakeup unit is used while the CPU waits for the current startup and LED voltage settlement time before starting the ADC acquisition. For a correct FOT implementation the ADC measures the V_{COMx} voltage when the PWM is ON and when the V_{LED} has a correct value. This time is fixed to around 100 µs after the SMED start. When the AWU time expires, the interrupt starts the ADC channel acquisition.

- `ADC_ISR()`: ADC interrupt

  The routine samples, for each channel, V_{LED_PW} and V_{COMx}, 4 times respectively. The values are then averaged and stored. The ADC sampling routine occurs 100 µs after the start of the PWM dimming. Should the PWM dimming time be smaller than 100 µs, the ADC samples incorrect voltage values and therefore the samples are discarded. This protection (inconsistent voltage protection) is implemented via the `SMED_Active` bit.

  The interrupt routine uses the `adc_eoc` flag to inform the main function that the ADC operation is completed and that new data is available. As a consequence, the software discards the ADC value. The interrupt routine uses the `adc_eoc` flag to inform the main function that the ADC operation is completed and that new data is available.

- `Get_Key()`

  This routine is used to detect if the board buttons are pressed. The STLUX385A uses the ADC (ADCIN[7], pin 31) to sense the button status. Since the ADC is used also to read V_{COMx} and V_{PWR}, a protection is set to avoid conflicts.

- `Comp_PWM()`

  It computes the new SMED's operating parameters as calculated by the adaptive voltage compensation mechanism.The new values are updated on the SMED registers only during the next STOP phases (using the `upd_flags` variable with `UPD_PWMx` flags).

- `Comp_DUTY()`

  This routine is invoked when the user changes the dimming period. The routine computes a new dimming period based on the `ontime[]` structures and stores the result on `tmp_onoff[][]` structures. The update is performed at the beginning of the next dimming period (using the `upd_flags` variable with `UPD_DUTY` flags) and when the dimming is OFF. During the update, the `tmp_off[][]` structure is copied into the `onoff[][]` structure.

The tasks displayed in the previous picture refer to the PWM and dimming control operations only. The STLUX385A firmware handles other tasks such as the command line, which uses two interrupts to read or write characters to the serial line. The serial interrupts use the lowest priority level among all the other sources (STMR, AWU, SMED and ADC).

5.3 Printf

The STLUX385A firmware implements the standard "`printf()`" subroutine, which takes several cycles to expand the printed strings and the operation delay real-time events. It is suggested the use of "`printf`" in production code is minimized or removed.

5.4 Low power modes

The STLUX385A may enter low power modes through the instruction WFI (wait for interrupt), where the internal CPU is stopped until an interrupt is received.

The "wi" console command instruction allows the user to use the WFI instruction (see *Section 7.4.10*).

The user may implement more aggressive low power schemes via new software routines. For example, the STLUX385A power consumption efficiency could be increased by disabling the SMEDs and the 96 Mhz PLL while the LED string is completely switched OFF.

6 Protection and regulation algorithm

This section describes the protection and regulation algorithms implemented in the STLUX385A. Every channel is independent from the others and the same logic applies to all channels.

The STLUX385A implements software and hardware protections aimed to preserve the external hardware in case of abnormal events or variations. The STLUX385A doesn't need external active analog circuitry on the board as the firmware replaces them. Once an error is recognized, the FW stores the error code and keeps sensing the system to verify that the problem has been fixed. The error code can be verified typing the "st" command (see *Section 7.4.6*). Protections can be summarized according to different priority levels.

6.1 Overcurrent protection

The overcurrent protection (OCP), which preserves the external circuit from damage due to a component fault, is the most important protection and requires a short detection and reaction time. The STLUX385A implements OCP directly in the SMED hardware (through states S1 and S2) and doesn't require any high priority firmware intervention.The internal SMED clock guarantees a detection time of ~10 ns (Fck_smed = 96 Mhz) however the external circuit response time pushes the overall detection time to 200 ns. When an OCP is detected, the SMED stops its PWM output and generates an interrupt. The firmware can then execute the recovery procedure by periodically restarting the PWM line.

6.2 Initial estimation

During the startup phase, the input power voltage is monitored by the Set_check_V() routine. This routine reads the power voltage (on ADC channel zero) and stores it into the global variable `Vpwr[]`. The voltage read back is stored into all channel locations (0 to 3). The routine Set_check_V() estimates the voltage drop generated by the LEDs by using the number of LEDs as specified by the user and the maximum and minimum predefined LED voltage drops. The following formula is used:

Equation 11

$$\frac{wk\_Volt[x].Min\_vl + wk\_Volt[x].Max\_vl}{2}$$

where x is the LED channel. The estimated voltage drop is used for the initial calculation of the T_{OFF} and T_{ON} values.

6.3 Adaptive voltage compensation

The adaptive compensation is a technique that monitors the system and changes the operational parameters to guarantee that the configured average current always flows through the LED. As the average current depends on the T_{ON} and T_{OFF} values, the STLUX385A recalculates the values at the beginning of every dimming ON cycles. The adaptive compensation is activated with the serial command "au x 1".

The Vpwr[x] and Vled[x] voltages are updated by the Check_V() routine on every dimming cycle. The voltages are read from the corresponding ADC channel 100 µs after the beginning of the dimming ON phase. This time offset guarantees that the LED are receiving the correct amount of current.

When the adaptive compensation is inactive, the Vpwr[x] and Vled[x] variables are not updated and store the voltages updated either during startup or before the compensation is disabled. When the adaptive compensation is inactive then the user can update the Vpwr[x] and Vled[x] variables via the serial line ("vp" and "vc" command). The Check_V() subroutine checks that:

- The PWM signal is OFF when the ADC completes the operation; all the ADC readings are discarded. This situation can happen during an OCP situation where the current peaks before the 100 µs limit.
- The Vpwr[x] voltage is lower than the vmax_50 constant (maximum power voltage). If it is high, the ERR_MAX_V error flag is set.
- The Vpwr[x] voltage is higher than the minimum imposed by the number of LEDs (wk_Volt[i].Min_vcc). Otherwise the ERR_MIN_L error flag is set.
- The common voltage point is higher than the minimum voltage imposed by the V_COM_MIN constant. The value of V_COM_MIN constant is 2.8 V. Otherwise the ERR_MIN_L error flag is set.
- The voltage across the LED is within the minimum (wk_Volt[i].Min_vl) and maximum (wk_Volt[i].Max_vl) values defined by the number of LEDs. Should the voltage across the LED be out of this range, then the ERR_MIN_L error flag is set.

a) The 100 µs delay is achieved using the auto-wakeup (AWU) feature provided by the STLUX385A.

b) Should SMED4 be available and not connected to a LED string, it can be used to generate the delay for the ADC measurement. This leaves AWU free for other purposes.

c) The two voltages Vpw[] and Vled[] are acquired on each dimming ON phase, four times alternately (CH0, CH1, CH0, CH1, CH0, CH1, CH0, CH1, CH0). The automatic ADC sequential measurement is a unique STLUX385A feature.

When the compensation is active, the Comp_PWM() routine calculates the T_{ON} and T_{OFF} based on the most recent voltage measurements. Should the resulting frequency be out of the designed range:

- The T_{ON} and T_{OFF} values are not updated.
- A single conservative T_{ON} - T_{OFF} period (T_{ON}[max]=3 µs, T_{OFF} = 5 µs) is applied during the actual dimming ON cycle.
- The error ERR_MAX_F or ERR_MIN_F is risen.
- The previous T_{ON} and T_{OFF} values are applied to the next dimming ON cycle.

From the application point of view, the LED frequency is bounded to this prevent excessive dissipations or audible noise.

6.4 Software crash protection

A final protection is used to prevent deadlock and software crashes and it's implemented via the watchdog timer (IWDG). The watchdog is reset in the main function and must be refreshed at least every 10.4 ms, otherwise the reset signal is internally generated to solve the lock-up situation. In the current software implementation, the "printf" routine also refreshes the IWDG timer due to the long time required to process strings.

7 User interface

7.1 Control buttons

The STEVAL385LED4CH demonstration board features three buttons, used to manually change the dimming duty cycle. The buttons can be configured to operate in two different ways and their behavior is controlled via the "ed" command. Holding a button causes its action to continue.

Single channel dimming: when the "ed" command is OFF ("ed 0"), the buttons behave as follows:

- "LED CH SEL": selects a channel. The new selected channel blinks once.
- "INC": increases the dimming duty cycle on the selected channel.
- "DEC": decreases the dimming duty cycle on the selected channel.

Multichannel dimming: when the "ed" command is ON ("ed 1"):

- The button "LED CH SEL" is not used.
- The button "INC" increases the dimming duty cycle on all channels.
- The button "DEC" decreases the dimming duty cycle on all channels.

The three buttons are read by the ADC channel line 7.

7.2 Status LEDs

There are two status LEDs attached to the board:

- Green LED (status): the green LED is active when the input power is correct and the STLUX385A is running. The LED can also act as a firmware performance monitor and show the CPU usage. The performance mode is enabled by the "vi" instruction (see *Section 7.3*) and the LED toggles as follows:
  – LED ON: CPU running
  – LED OFF: CPU sleeping
- Red LED (failure): the red LED is used to report an error or the intervention of a protection to preserve the hardware. Once the red LED is ON, the only way to clear is to use the "co" command (see *Section 7.3*).

7.3 Serial interface

This section describes the terminal interface line available on the STLUX385A firmware. The serial line allows the configuration of the:

- Working point of the LED channel
- Current level
- LED numbers
- Dimming percentage

It also allows the device internal status to be inquired.

The STLUX385A command line interface is accessible via a standard terminal utility installed on the user's PC.

The serial (TTL 0 - 3.3 V) cable is shipped together with the demonstration board. The on board connector is identified as UART I/F. The serial line uses only three lines: GND, TX, RX, the baud rate is 115200, no parity, 8-bit, 1 stop bit, no handshake. The user should identify the PC COM line number, please refer to your local IT department if the COM number is difficult to retrieve. The PC also installs the serial COM drivers as soon as the cable is connected. If the driver is not installed, please call your IT support.

Warning: The command line executes all the commands entered. The user has the responsibility to use the correct commands as per circuitry adopted. For example, should the user configure 1 A current through a 300 mA LED chain, the LEDs could be damaged.

The STLUX385A, during power-up and reset, sends the following strings to the serial line. The strings are received and displayed on the console:

Figure 9. Console screen during startup

```
EVALSTLUX385 - 4chLED  VxRx
Build  Jan  20  2012  14 : 00 : 44
Ready
```

AM16709v1

This message is used to verify that the link between the PC and the STLUX385A is up and that the correct baud-rate is selected. The strings "VxRx" report the firmware version and revision and, on the next line, the time and the date during the compilation process. "Ready" appears when the STLUX385A chip is ready to receive a command.

At this point, the user can interact with the STLUX385A using one of the following commands.

Note: The command list is shown by the "?" command.

Figure 10. “?” command

```
Monitor channel command
lc   [ch]   [I]             Setup led current [0 - 10] for channel [ch]
ll   [ch]   [0; 6 - 256]    Setup led dimming for channel [ch]
ln   [ch]   [num]           Setup led numbers [3- 10] for channel [ch]
ad  [ch]                    Get adc channel [0- 7] value
au  [ch]    [0,1]           Control [ch] loop, 1=en. 0=dis
st                          Get status
pw [ch]                     Get pwm status on    channel [0   - 3]
vp  [ch]    [0 - 1023]      Simul. POW voltage on channel [0 - 3]
vc  [ch]    [0 - 1023]      Simul. Common voltage on channel  [0 - 3]
wi  [0,1]                   Enable [1], disable [0] the wfi instruction
dm [add]  [len]             dump register/memory
rr   [add]                  get register/memory
rw  [add]  [dat]            write register/memory
ed  [0,1]                   Enable [1] disable [0] global dimming key/cmd
di   [value]                Globally dimming the light to [0- 100]%
ti                          Get time
co                          Clear OVC flag
hl  [??]                    help cmd [??] string
hl                          get help
?                           get help
use [enter] to end line,    use [backspace] to delete char
                                                               AM16710v1
```

Every command is composed of two parts: the command prefix (normally two characters) and the command fields. Any incorrect typed character can be changed using the backspace character.

Each LED channel is identified by the channel (ch) number only (from 0 to 3).

7.4 Commands

The following section describes the commands accessible through the console.

7.4.1 lc - change LED current

The command is used to change the current in the specific LED chain. The configured value is stored into the E^2PROM data area for its read-back and applies to next power-on.

Syntax: lc [ch] [I]

Parameters:

- ch [0-3]: the LED string channel value.
- I [0-10]: current index as shown in *Table 1*. It represents the peak of the chain current and the index ranges from 0 (273 mA) to 10 (1184 mA).

Output: none

Example: lc 2 3: configures channel 2 to use 492 mA mean current.

7.4.2 ll - change LED dimming level

To change the LED dimming level, use the "ll" command. The value is stored into the E^2PROM data area for its read-back and applies to next power-on.

Syntax: ll [ch] [level]

Parameters:

- ch [0-3]: the LED string channel value.
- level: [0-256]: indicates the dimming level in a scale from 0 (OFF) to 256 (always ON), a value between 1 (0.4%) and 255 (99.6%) indicates the dimming level. A value of 256 turns the LED chain on at full brightness (100%). When the global dimming level is different from 100% (via "ed" and "di" command), the equivalent current on example is equal to ('level / 256)*(global dimming level).

Output: none

Example: "ll 3 200": sets up channel 3 to use a current equal to 200/256 of the maximum current defined by the "lc" command.

7.4.3 ln - configure LED number

Used to setup, before connecting any LED, the number of LEDs used in a specified chain. The value is stored into the E^2PROM data area for its read back and applies to next power-on.

Syntax: ln [ch] [num]

Parameters:

- ch [0-3]: the channel number.
- num [3-10] the number of LEDs. The minimum LED number is 3; the maximum LED number is 10.

Output: none

Example: "ln 1 6": sets up six LEDs on the channel 1.

7.4.4 ad - read ADC channel

Reads the ADC channel [ch] and displays the result in decimal value.

Syntax: ad [ch]

Parameters:

- ch [0-7]: ADC channel number. For the ADC channel association, see the board schematics.

Output:

- The output value is the decimal conversion read from the ADC channel. The value read from the LED channel (V_{COMx} point) depends on the dimming level. The correct input voltage at the STLUX385A pin is given by the ADC value multiplied by 0,001221896 [= (1.25/((210)-1))]. For channels 0, 1, 2, 3, 4, 6, the ADC value has to be multiplied for 0.05437489.

Example: "ad 0": reads the power line voltage.

7.4.5 au - adaptive voltage compensation

The adaptive voltage compensation (control loop) is used to calculate the new PWM parameters in relation to the voltage read by the V_{COMx} or V_{LED_PW}. When the adaptive voltage compensation is OFF ("au x 0"), the voltage compensation simulation mode is enabled and the commands vp and vc become available. The value is stored into the E^2PROM data area and applies to next power-on.

Syntax: au [ch] [enable]

Parameters:

- ch [0-3]: channel number
- enable [0,1]: enables (1) or disables (0) the adaptive voltage compensation.

Output: none

7.4.6 st - verify status

The command reports the STLUX385A status.

Syntax: st

Parameters: none

Output: an example of the output is shown:

Figure 11. "st" command

```
Status: err=0   cnt=0     di=1:050
Led ch=0 on  l=1   d=010  led=6  cur=0   Vpw=432 Vcom=80 OVC=off
Led ch=1 on  l=1   d=020  led=6  cur=3   Vpw=432 Vcom=89 OVC=off
Led ch=2 on  l=1   d=210  led=6  cur=2   Vpw=429 Vcom=91 OVC=off
Led ch=3 on  l=1   d=055  led=6  cur=1   Vpw=430 Vcom=94 OVC=off
```

AM16711v1

The first line reports the last error [err=] and the total number of errors found [cnt=] by the FW. The [di=] field reports the "ed" status [0 or 1] and the "di" global dimming percentage (from 0% to 100%). It also reports, for each channel [ch], the channel activation or deactivation [on or off], the automatic loop status [l=1 or l=0], the dimming level [d=010], the number of LEDs, the current index and the V_{LED_PW} and V_{COMx} voltages when the dimming is ON. The voltage data (V_{LED_PW} and V_{COMx}) is the raw data read from the ADC. Multiply by 0.05437489 to convert into a voltage.

7.4.7 pw - PWM status

Syntax: pw [ch]

Parameters:

- ch [0-3]: channel number

Output:

- Retrieves the TIM_Cnt values configured in the [ch] channel. The command "pw 0" output is shown in *Figure 12*:

Figure 12. "pw" command

```
Led ch=0 on S0=156 S1=276 S2=939 D=32
```
AM16712v1

It reports the channel status (ON or OFF) and the TIM_Cnt counters for states S0, S1 and S2, frozen at the moment when the command is invoked. The counters in S0, S1 and S2 are based on the 964 MHz clock unit. The command also reports the dimming period ("D=32"). In this case, the value is the dimming period and starts from 1 (0.4%) and ends to 256 (100%).

7.4.8 vp - set V_{LED_PW} simulated voltage

The command is used during the voltage compensation simulation phase ("au ch 1") to change the simulated voltage read from the V_{LED_PW} ADC pin. This command is enabled only when the adaptive voltage compensation is disabled ("au ch 0" command).

Syntax: vp [ch] [voltage]

Parameters:

- ch [0-3]: channel number
- the parameter represents the new simulated voltage value, expressed as (real voltage / 0.05437489)

Output: none

Example: "vp 0 590": simulates the power voltage at 32.08 volt.

7.4.9 vc - set V_{COMx} simulated voltage

The command is used during the voltage compensation simulation phase ("au ch 1") to change the simulated voltage read from the V_{COMx} ADC pin. This command is enabled only when the adaptive voltage compensation is disabled ("au ch 0" command).

Syntax: vc [ch] [voltage]

Parameters:

- ch [0-3]: channel number
- voltage [0-1023]: same as the "vp" command

Output: none

7.4.10 wi - CPU load monitor and low power mode

The command enables the CPU load monitor where the CPU load can be observed via the green LED on the demonstration board. The command also enables the STLUX385A low power mode by using the WFI (wait for interrupt) instruction when idle. The "wi" command is used to reduce the power consumption of the STLUX385A.

7.4.11 dm - dump memory

It displays the memory content of a specific area of memory.

Syntax: dm [add] [len]

Parameters:

- add: the memory address. It is possible to use a hexadecimal value using the "0x" prefix. The total memory area available on the STLUX385A is 64 Kbyte.
- len: total length of memory to be retrieved, in bytes. "0x" can be used as prefix for a hexadecimal number.

Output:

- The memory content starting from [add] address and [len] bytes long.

7.4.12 rr - read memory, single byte

The command reads and displays the specified byte of memory.

Syntax: rr [add]

Parameters:

- add: the memory address. It is possible to use a hexadecimal value using the "0x" prefix. The total memory area available on the STLUX385A is 64 Kbyte.

Output: the byte located at [add] address

7.4.13 rw - write memory, single byte

The command writes a single byte of memory. The incorrect use of this command could overwrite internal registers and change the behavior of the device and could cause the destruction of the demonstration board.

Syntax: rw [add] [dat]

Parameters:

- add: the memory location to write. It is possible to use a hexadecimal value using the "0x" prefix. The total memory area available on the STLUX385A is 64 Kbyte.
- dat: the byte to be written. It is possible to use a hexadecimal value using the "0x" prefix.

Output: none

7.4.14 ed - enable global dimming function

It enables or disables the global dimming function. When the function is enabled, the "di" command applies to all channels. At the same time, the "+" and "-" buttons operate in global dimming mode. When the function is disabled, the "di" command is disabled and the LED dimming can only be changed via the "ll" command. Besides, the "+" and "-" buttons control a single channel. The [ed] value is stored into the E^2PROM data area for its read back at the next power-on. On reset, should "ed" be enabled, the last "ll" dimming value is not applied immediately but with a gradual light increment (from 0% to the "ll" value).

Syntax: ed [enable]

Parameters:

- enable [0,1]: enables or disables the global dimming function.

Output: none

7.4.15 di - set global dimming value

It sets the dimming of the LED strings to be used when the board is operating in global dimming mode ("ed 1"). At power-on the "di" value applied is 100 (equal to 100%) after a ramp-up period only if "ed" command is enabled.

Syntax: di [value]

Parameters:

- value [0,100]: the percentage of the value configured with the "ll" command. The global dimming is applied to all channels.

Output: none

Example: "ll 0 220", "di 50". Channel 0 is dimmed at 220 @ 50% = 110. All other channels are dimmed at 50% of the their current dimming value.

7.4.16 ti - read time counter

It retrieves the time since the last power-up or reset.

Syntax: ti

Parameters: none

Figure 13. Console screen for "ti" command

```
Time is 0x00000098: 7d
```

AM16713v1

Output: the time counter, in hexadecimal.

The output is composed of two parts, separated by the comma sign. The first part shows the number of seconds elapsed since power-on or reset. The second part is the tick: a number from 0 to 191 (in hex: 0x00 to 0xBF) that represents the number of dimming period elapsed. One dimming period is exactly 5.1813 ms (193 Hz).

7.4.17 co - clear error

It clears any system error. This command clears ("off") the error code, removes the OCP flag monitored by the "st" command and switches OFF the fault LED. The command does not clear the error counter tracked by the "st" command. The fault LED may remain switched ON after the "co" command if the cause of error has not been removed.

Syntax: co

Parameters: none

7.4.18 hl - command help

It visualizes a brief description of the command selected.

Syntax: hl [command]

Parameters:

- command: a terminal command

Output: the description of the command selected.

7.4.19 hl - ? - global help

It displays a list of all accepted commands.

Syntax:

- hl
- ?

Parameter: none

Output: the list of all accepted commands.

7.5 Error codes

The STLUX385A firmware performs several system checks during its execution. When an error is found:

1. The red fault LED is triggered.
2. The error counter is incremented.
3. The error code is stored in an internal variable. The last error code is reported by the status command ("st").

The fault LED can only be reset via the "co" command. The "co" command also clears the error code. The error counter is never reset.

The following table describes the error codes:

Table 3. Error codes

Code	Mnemonic	Meaning	Action
0	ERR_NO_ERR	No error	N/A
1	ERR_PW	Power voltage under or upper the limit at power-on	Dimming OFF
2	ERR_MAX_F	PWM frequency is over the limit	N/A
3	ERR_MIN_F	PWM frequency is under the limit	N/A
4	ERR_FLASH	Error while writing in Flash	The last serial command is not stored into E^2PROM
5	ERR_OVC	Overcurrent protection (OCP). Preventing short-circuit or overcurrent point	Dimming OFF
6	ERR_MAX_V	Main input voltage over the upper limit (50 V)	v
7	ERR_MIN_V	Main input voltage too low to drive the configured number of LEDs.	N/A
8	ERR_MIN_L	Power voltage across the inductance lower the limit (lower than 2.8 V)	Dimming OFF
9	ERR_MAX_L	LED voltage over the maximum.	N/A
10	ERR_MIN_D	Number of LEDs below the limit (< 3 LEDs)	N/A
11	ERR_VOL_L	LED voltage drop below the minimum	N/A
12	ERR_DALI	Failure on the DALI I/F – only when DALI FW is enabled	DALI error

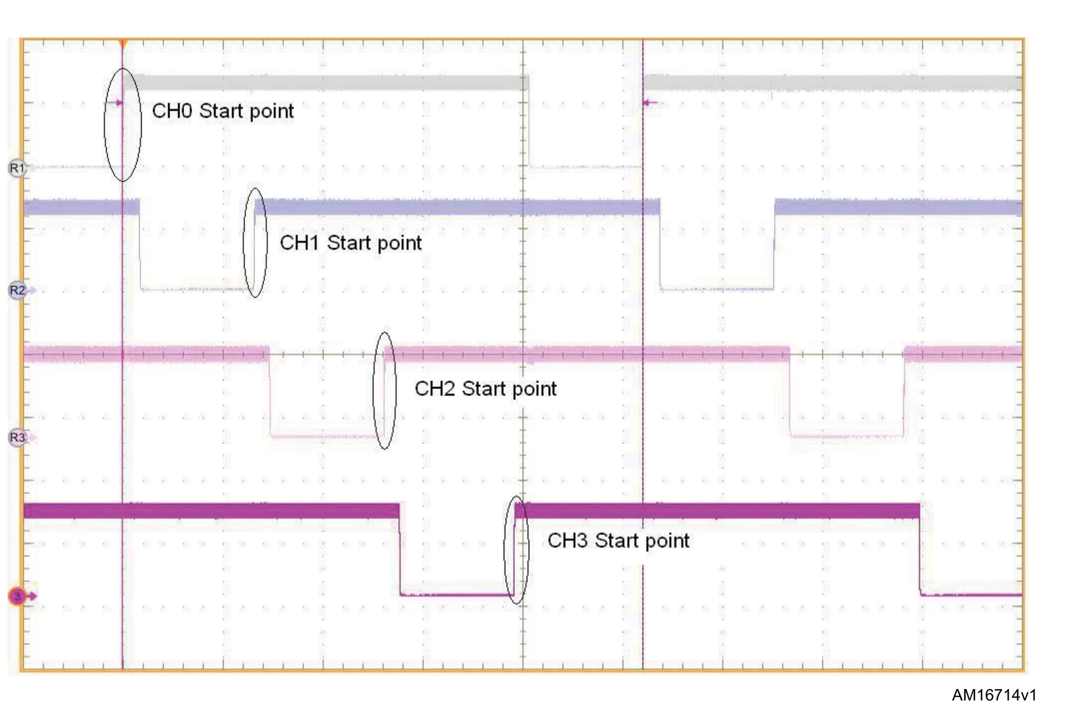
8 Measurements

This section reports several measurement results validating the STEVAL385LED4CH demonstration board performance.

8.1 Output current precision for different current setup

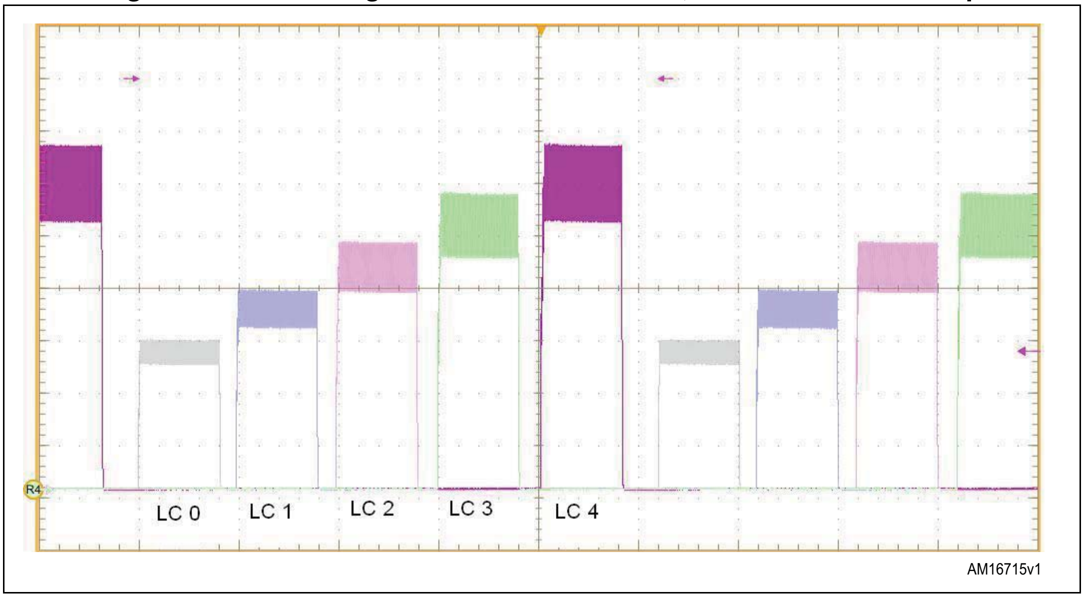
Figure 14 shows the current flow through the LEDs, measured with a current probe on all four channels. The operating current of all channels is the point 0 in *Table 1* (245 mA mean). Note the channel has delayed startup point and the current precision on all channels displayed. R1 is channel zero, R2 is channel two, R3 is channel three and 3 is channel four. The dimming periods, same for all channels, is ~5 ms (193 Hz).

Figure 14. Current regulation - 4 channels at 245 mA



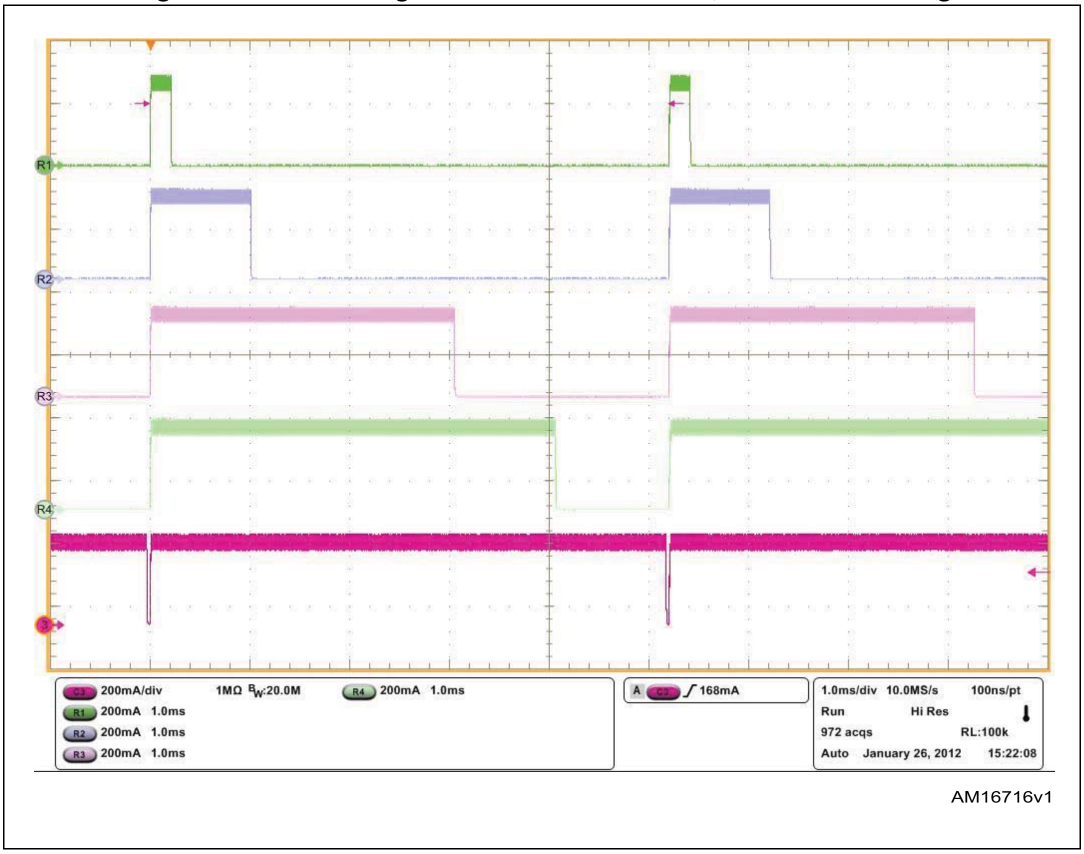
The following picture represents the different current regulation on the same channel and at the same dimming level. The current has changed using the "Ic" command using, as second parameter, the index of the expected current. The vertical grid resolution is 100 mA/div. See *Table 1* for the index and current relation.

Figure 15. Current regulation - same channels, different current setup



Next picture shows a single channel with different dimming levels. The dimming values are: R1 = 20 units (7.8%); R2 = 50 units (19.5%); R3 = 150 units (58.6%); R4 = 200 units (78.1%); 3 = 255 units (99.6%). The OFF phase of the 3 line is equal to one dimming unit corresponding to 1/256 parts of 5,12 ms (~20 µs).

Figure 16. Current regulation - same channels, different dimming



8.2 Efficiency according to different current setup

The following table shows the efficiency reached based on different number of LEDs. All 4 channels are using the same current. The measure reports the minimum, average and maximum efficiency for a configured fixed current on all input power voltage range. The current is calculated based on *Table 1*.

The plots use "y" axis representing the efficiency, expressed as a percentage.

The board configuration is with the LED always ON ("ll x 256" and, if "ed" is enabled, "di 100").

The best efficiency value measure is 98%. Since the board is designed to be "generic" and supports a wide range of configurations (LED numbers, power voltage range, etc.) the efficiency reaches 85% in some cases.

The input board power when the PWM outputs are disabled (the LED strings are OFF) is ~300 mW@12 V and ~600 mW@48 V.

Table 4. Efficiency according to different current when the input voltage changes

Plot index	Average current	Efficiency								
		3 LEDs			6 LEDs			10 LEDs		
		min.	mean	max.	min.	mean	max.	min.	mean	max.
1	**245 mA** (*Ic x 0*)	92.8	93.4	94.5	92.5	94.2	95.8	96.0	97.3	98.5
2	**329 mA** (*Ic x 1*)	92.1	92.6	94.1	93.7	95.0	96.5	95.5	96.0	96.5
3	**492 mA** (*Ic x 3*)	91.1	91.6	92.1	94.4	95.1	96.1	95.9	96.0	96.2
4	**738 mA** (*Ic x 6*)	89.2	89.5	89.9	94.1	94.8	95.2	95.3	95.4	95.5
5	**1.065A** (*Ic x 10*)	84.9	85.1	85.4	92.3	92.8	93.1	93.4	93.9	94.1

Figure 17. String efficiency based on different current levels - 3 LEDs for each string

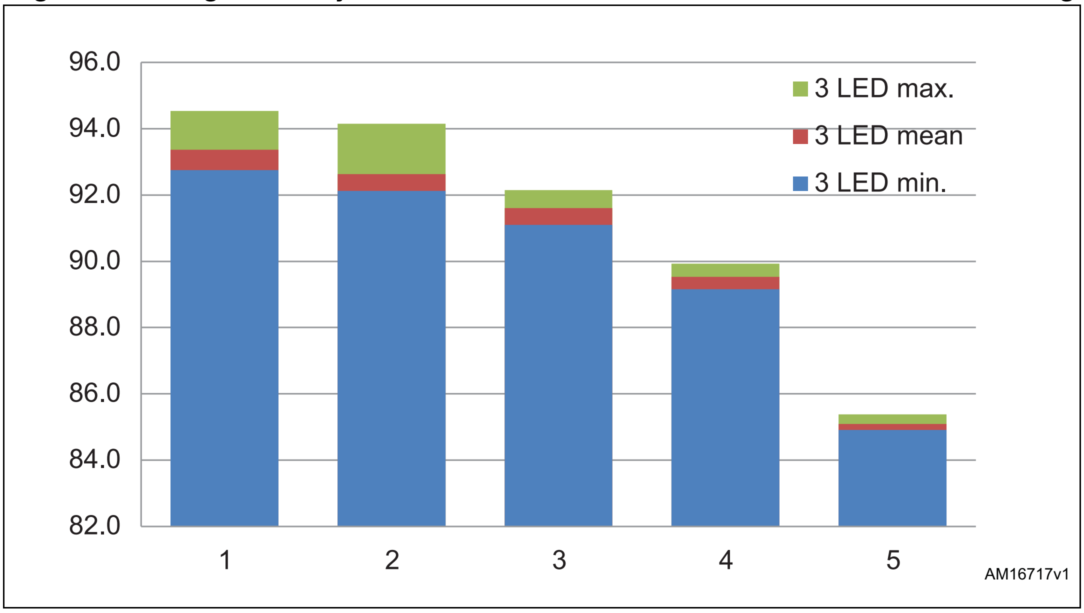


Figure 18. String efficiency based on different current levels - 6 LEDs for each string

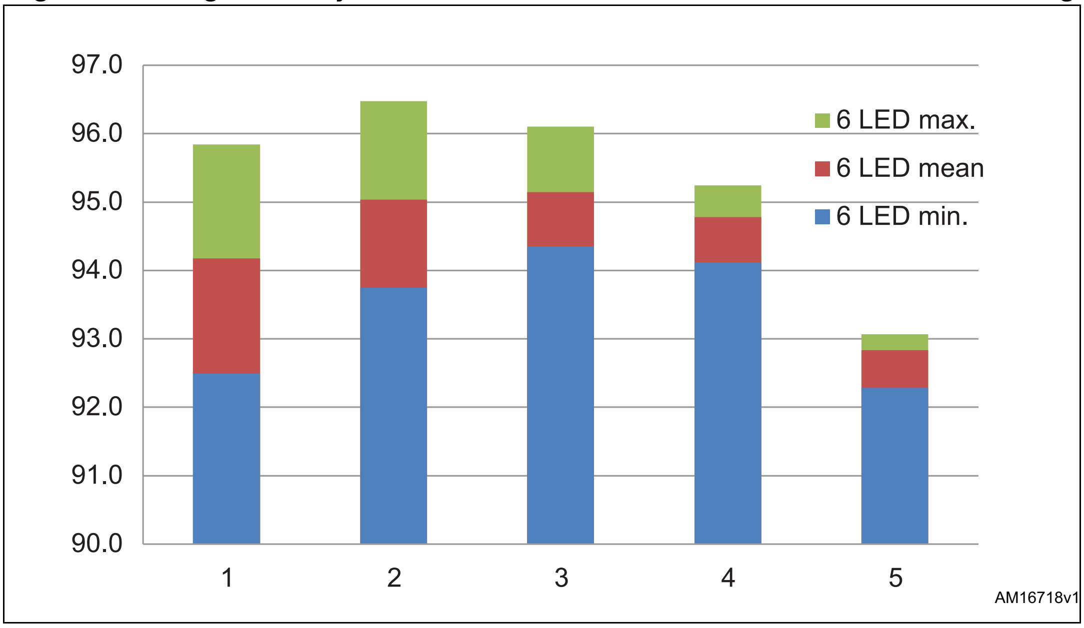
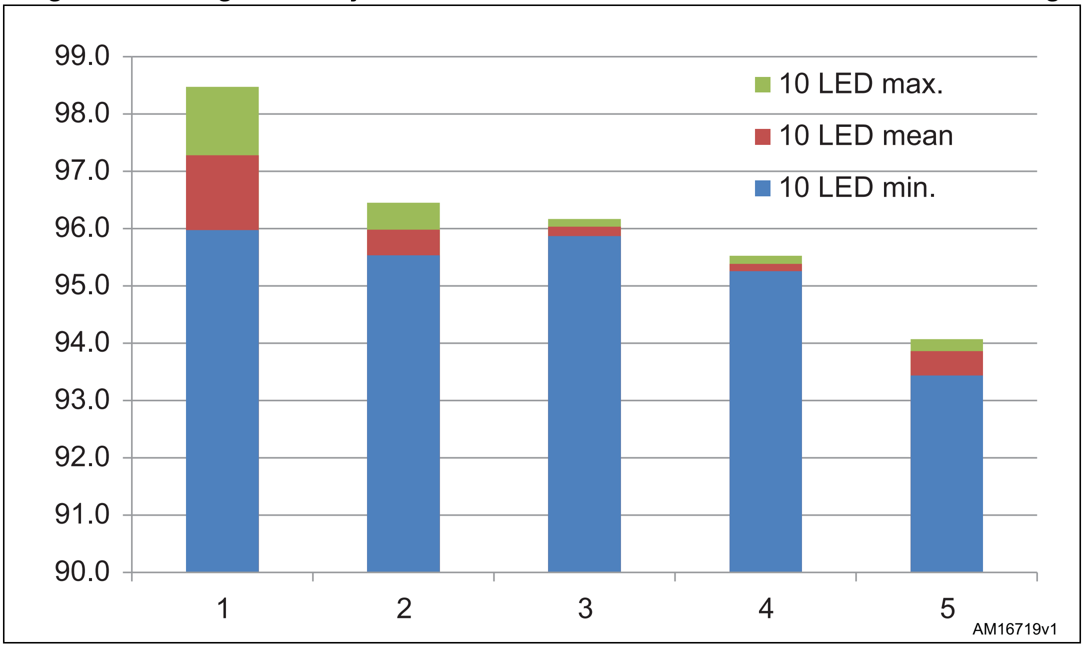


Figure 19. String efficiency based on different current levels - 10 LEDs each string



8.3 Current regulation on input voltage

The following graphs show the current regulation in relation to the input power voltage. The plots report the current on the "y" axis and the voltage on the "x" axis. The input power voltage in every figure is the effective voltage range only for the number of LEDs selected.

Figure 20. Current regulation vs. V_{PW_LED}: 3 LEDs

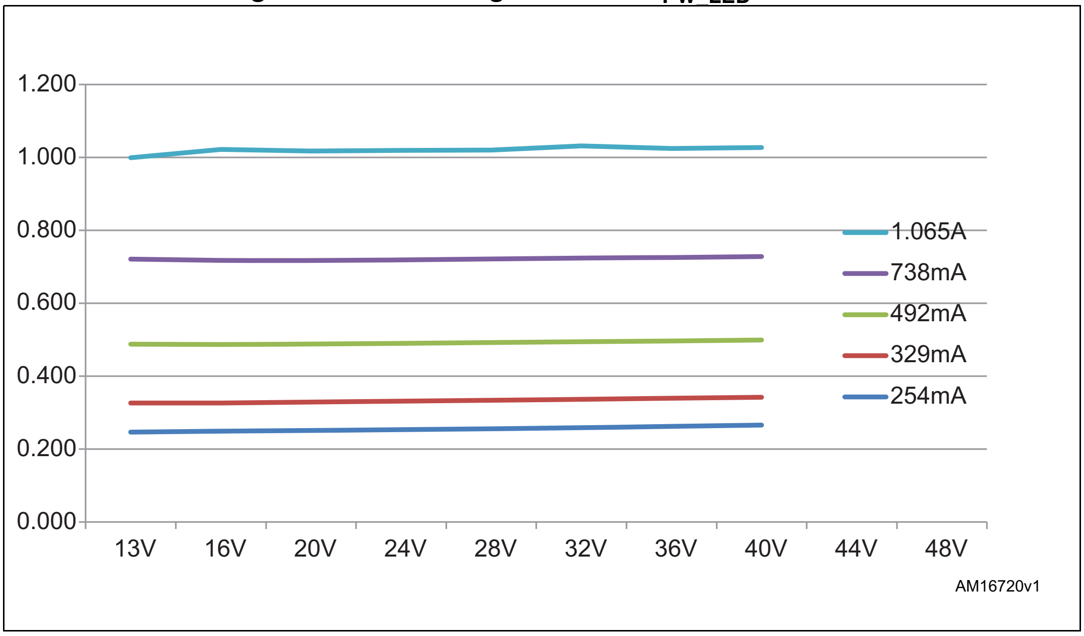


Figure 21. Current regulation vs. V_{PW_LED}: 6 LEDs

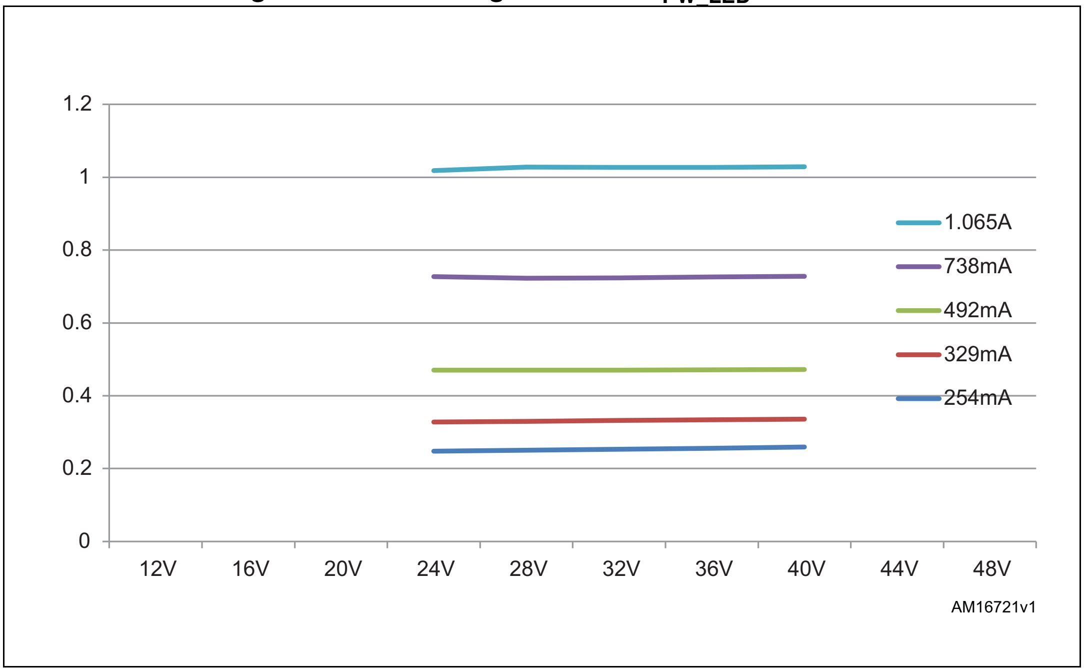
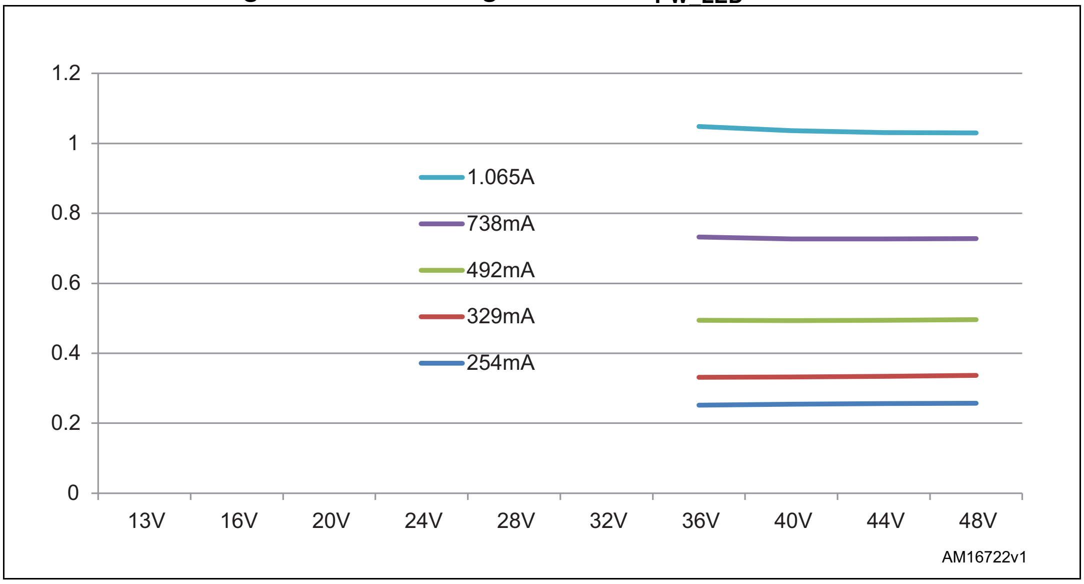


Figure 22. Current regulation vs. V_{PW_LED}: 10 LEDs



8.4 Current regulation validation

The *Table 5* compares the real current values, as captured by the oscilloscope, with the expected ones as configured by the user. The input power voltage is fixed and set at the middle of the usable range. The current is the mathematical average current (not the peak current) as got and calculated by the scope.

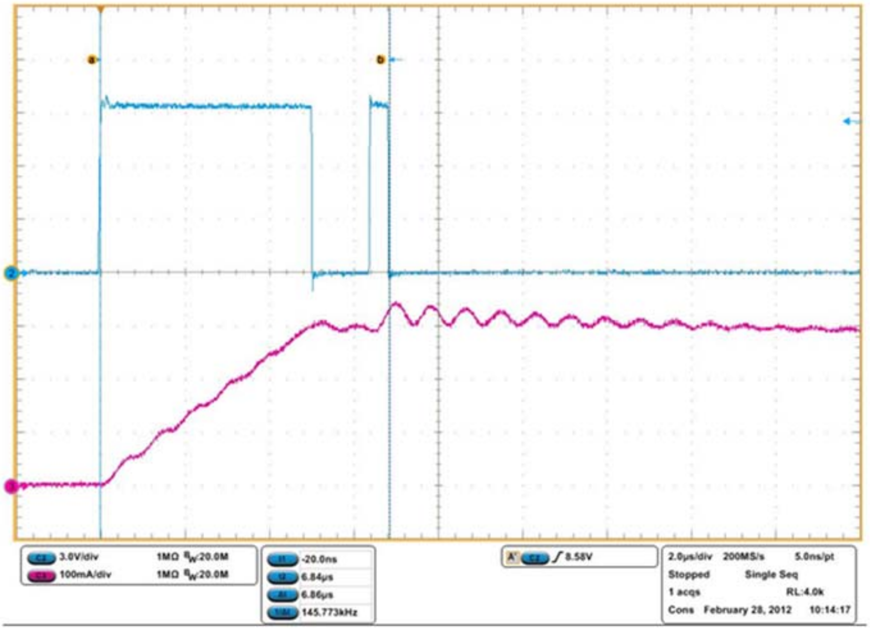
Table 5. Current regulation discrepancies

Command	3 LEDs – 20 V	6 LEDs – 32 V	10 LEDs – 44 V	Expected
Ic x 0	0.251 A	0.254 A	0.255 A	0.245 A
Ic x 1	0.329 A	0.333 A	0.333 A	0.329 A
Ic x 2	0.407 A	0.414 A	0.413 A	0.410 A
Ic x 3	0.486 A	0.493 A	0.492 A	0.492 A
Ic x 4	0.563 A	0.571 A	0.570 A	0.574 A
Ic x 5	0.639 A	0.647 A	0.646 A	0.648 A
Ic x 6	0.715 A	0.725 A	0.724 A	0.738 A
Ic x 7	0.791 A	0.802 A	0.800 A	0.819 A
Ic x 8	0.867 A	0.879 A	0.877 A	0.901 A
Ic x 9	0.943 A	0.957 A	0.955 A	0.984 A
Ic x 10	1.016 A	1.031 A	1.030 A	1.065 A

8.5 OCP protection time

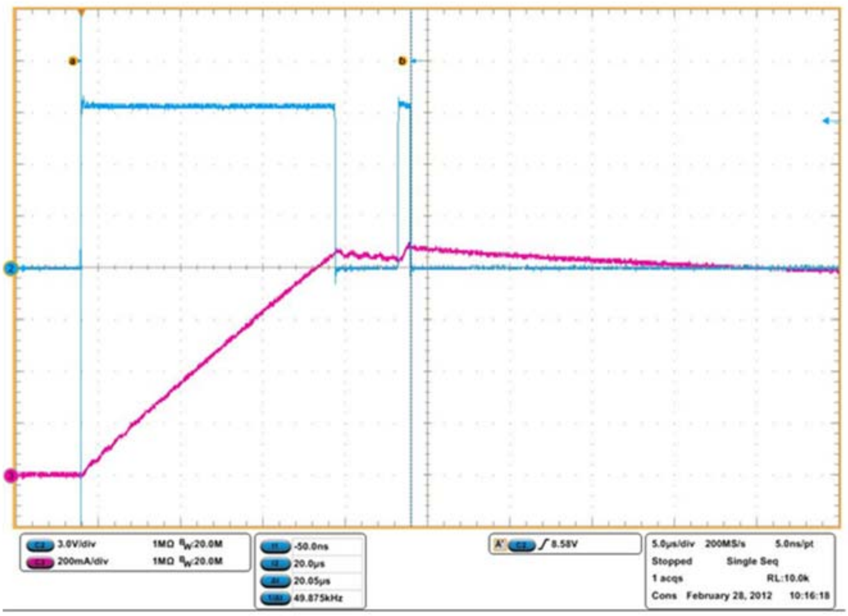
Figure 23, *Figure 24* and *Figure 25* show the current protection reaction time. The current working point is fixed at 245 mA in the first plot, at 738 mA in the second and at the maximum current (1.065 A) in the third. In all pictures, the blue track (2) shows the MOSFET gate signal while the purple track (3) displays the output current when the output is shorted. The cycle starts every 5.12 ms.

Figure 23. MOSFET gate during short-circuit - 245 mA



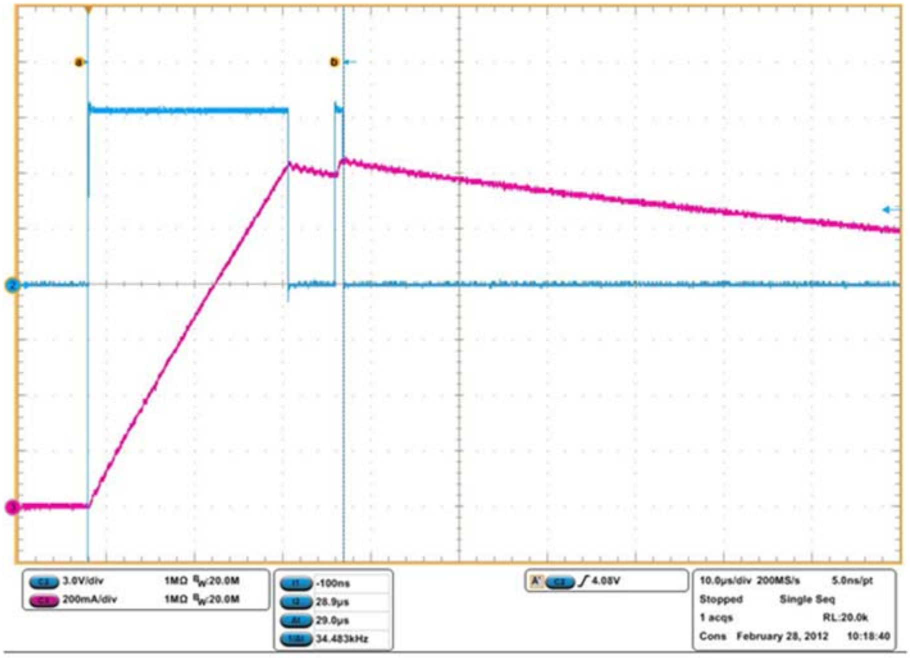
AM16723v1

Figure 24. MOSFET gate during short-circuit - 738 mA



AM16724v1

Figure 25. MOSFET gate during short-circuit - 1.065 A



AM16725v1

9 Pinout

Table 6 describes the STLUX385A pinout and the associated function.

Table 6. STLUX385A pinout

Pin	Dir.	Function	Note
1	OUT	PWM[0]	PWM - LED chain 0
2	OUT/IN	DIGIN[0]/CCO_clk	Internal clock check output – (option)
3	IN	DIGIN[1]	DIG1
4	OUT	PWM[1]	PWM - LED chain 1
5	OUT	PWM[2]	PWM - LED chain 2
6	IN	DIGIN[2]	DIG0
7	IN	DIGIN[3]	DIG3
8	OUT	PWM[5]	Output to R/C – voltage reference on CPM3 (LED0) – (optional)
9	I/O	SWIM	Debug interfaces
10	I/O	RESETn	To reset key and debug I/F
11	PS	VDD	Power to 3.3 V or 5 V
12	PS	VSS	Digital GND
13	PS	VOUT	Filter capacitance for 1.8 V internal voltage
14	OUT	LED (red)	Fault signaling or DALI TX line
15	OUT	LED (green)	Normal signaling (blink) or DALI RX line
16	OUT	PWM[4]	Option – PWM for the DC-DC converter
17	IN	DIGIN[4]	DIG2
18	IN	DIGIN[5]	Not used or DALI key input (option)
19	OUT	PWM[3]	PWM - LED chain 3
20		OSC_out	External oscillator or FW trigger used to debug application
21		OSC_in	External oscillator or FW trigger used to debug application
22	OUT	UART TX	Serial TX line
23	IN	UART RX	Serial RX line
24	A-IN	CPP3	Compare 3 – LED chain 0
25	A-IN	CPP2	Compare 2 – LED chain 1
26	A-IN	CPM3	Connect to PWM[5] – chain 0 current (5 mA/step) – (option)
27	A-IN	CPP1	Compare 1 – LED chain 3
28	A-IN	CPP0	Compare 0 – LED chain 2

Table 6. STLUX385A pinout (continued)

Pin	Dir.	Function	Note
29	PS	VDDA	Analog power supply
30	PS	VSSA	Analog ground
31	A-IN	ADCIN[7]	Key input – voltage depends whether key is pressed or not
32	A-IN	ADCIN[6]	External dimming signal – 0 -10 V equal to 0% 100% – (option)
33	A-IN	ADCIN[5]	Not used
34	A-IN	ADCIN[4]	Analog input CH4 – V_{COM3} – voltage from LED3 chain
35	A-IN	ADCIN[3]	Analog input CH3 – V_{COM2} – voltage from LED2 chain
36	A-IN	ADCIN[2]	Analog input CH2 – V_{COM1} – voltage from LED1 chain
37	A-IN	ADCIN[1]	Analog input CH1 – V_{COM0} – voltage from LED0 chain
38	A-IN	ADCIN[0]	Analog input CH0 – ADCIN0 – voltage on V_{LED_PW} signal (V_{IN})

Note: When the DALI I/F is used, please remove R78, R77 resistor (lose RUN-FAULT function), enable the "USE_DALI" constant on the "led.h" source file, recompile and load the new code.To request the source code for the DALI stack, please contact your local ST representative. The STEVAL-ILM001V1 kit gives access to the DALI hardware interface.

10 Schematic diagram

Figure 26. Schematic - STLUX385A - top

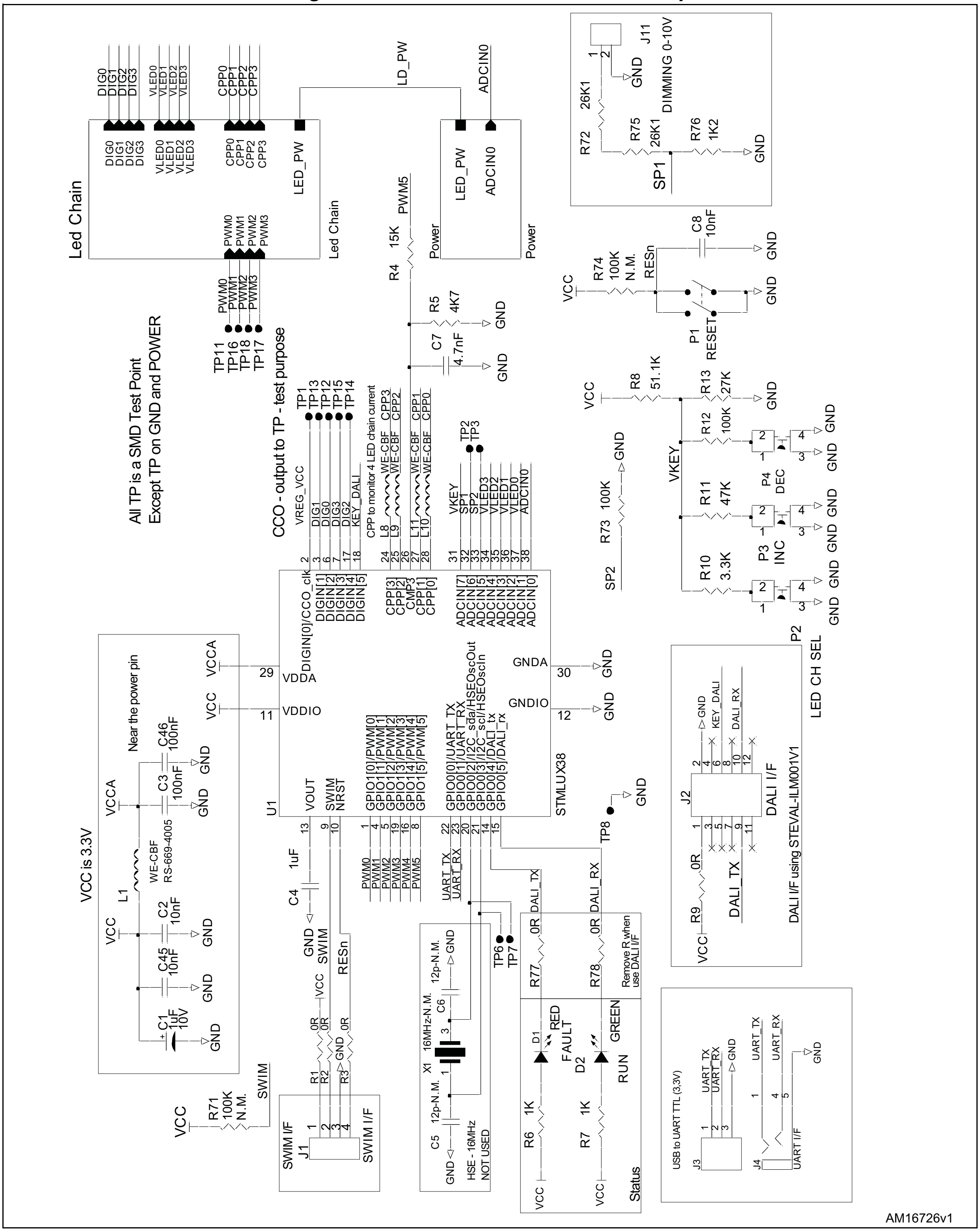


Figure 27. Schematic - 4 channel LED driver

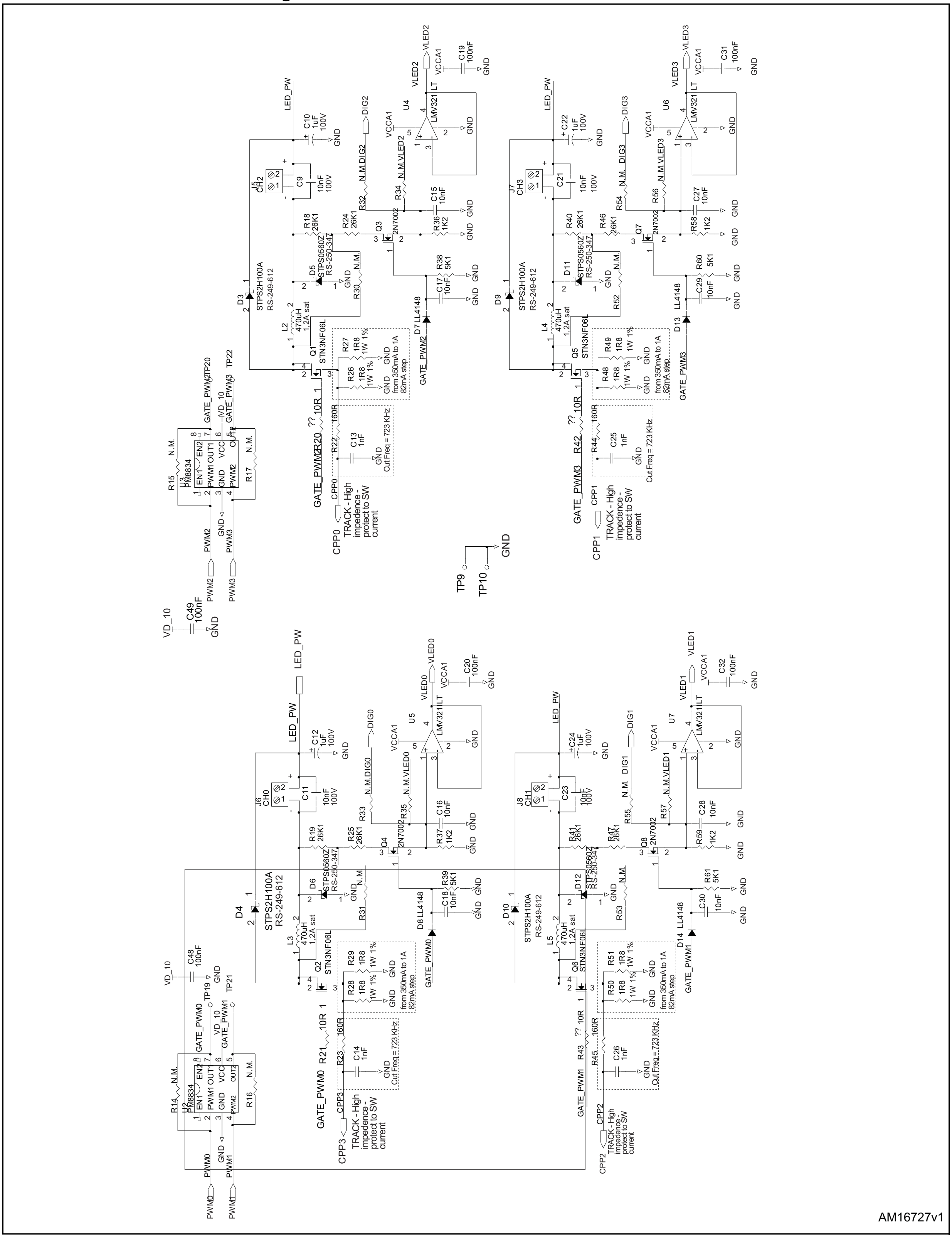
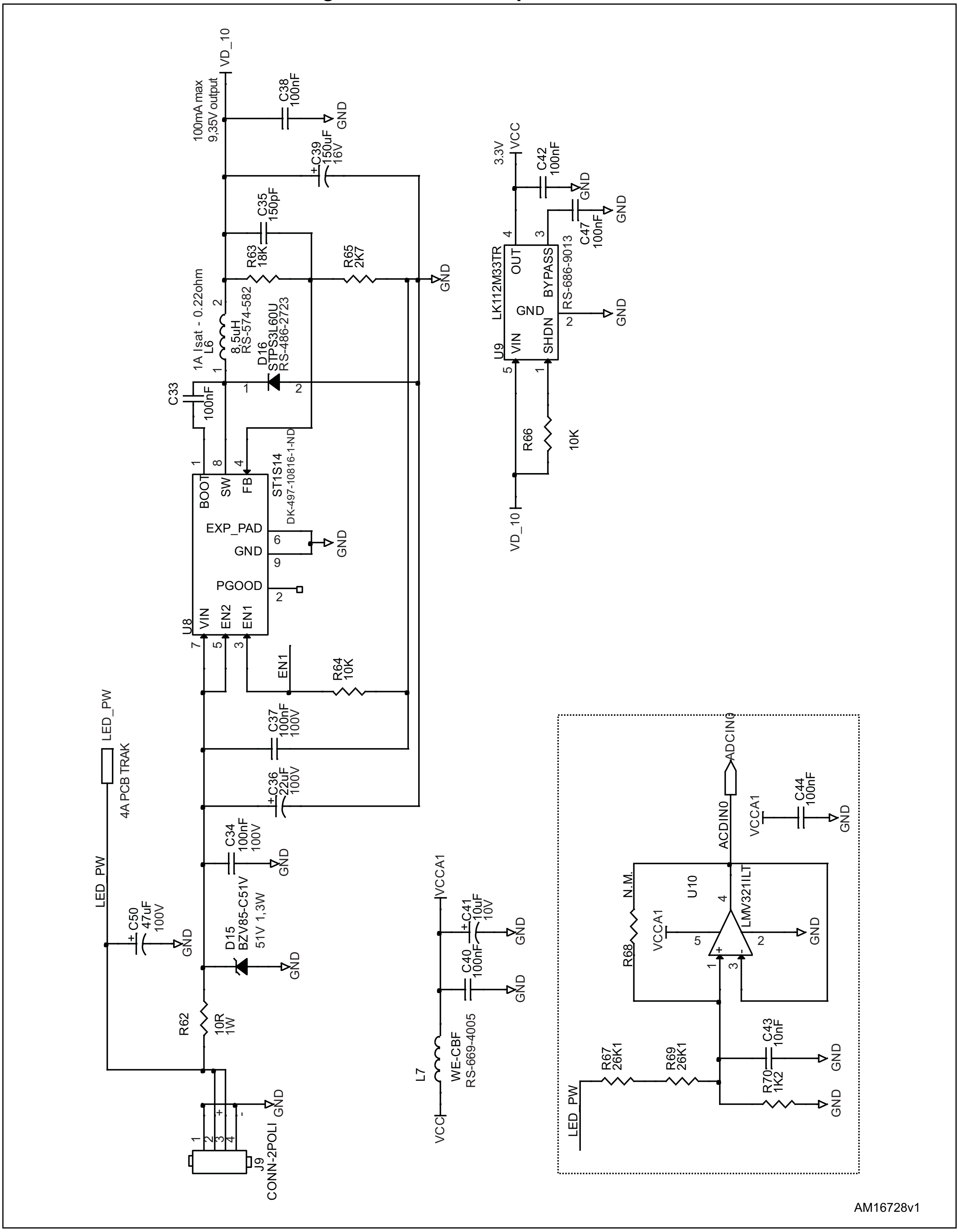


Figure 28. Schematic power section



11 Bill of material

Table 7. Bill of material

Item	Quantity	References	Part value	Package and supplier code	Notes
1	2	C1,C4	1 µF	Capc-0603	10 V
2	4	C2,C8,C43,C45	10 nF	Capc-0603	
3	8	C3,C19,C20,C31,C32, C40,	100 nF	Capc-0603	
		C44,C46			
4	2	C5,C6	12p-N.M.	Capc-0603	Not mounted
5	1	C7	4.7 nF	Capc-0603	
6	4	C9,C11,C21,C23	10 nF	Capc-1206	100 V
7	4	C10,C12,C22,C24	1 µF	Capc-1206	100 V
8	4	C13,C14,C25,C26	1 nF	Capc-0603	
9	8	C15,C16,C17,C18,C27, C28,C29,C30	10 nF	Capc-0603	
10	6	C33,C38,C42,C47,C48, C49	100 nF	Capc-0603	25 V
11	2	C34,C37	100 nF	Capc-1206	100 V
12	1	C35	150 pF	Capc-0805	25 V
13	1	C36	22 µF	D250P100	100 V
14	1	C39	150 µF	Capc-7343	16 V
15	1	C41	10 µF	V-3216	10 V
16	1	C50	47µF	Cap-axP1000D400	100 V
17	1	D1	Fault	Ledc-0603	
18	1	D2	Run	Ledc-0603	
19	4	D3,D4,D9,D10	STPS2H100A	SMA	
20	4	D5,D6,D11,D12	STPS0560Z	Sod123	
21	4	D7,D8,D13,D14	LL4148	Sod80-st	
22	1	D15	BZV85-C51V	DO41	51 V 1,3 W
23	1	D16	STPS3L60U	SMB	
24	1	J1	SWIM I/F	4 pin-pitch 2.54	
25	1	J2	DALI I/F	Pin 2X6 pitch 2.54	
26	1	J3	UART I/F - TTL	3 pin-pitch 2.54	

Table 7. Bill of material (continued)

Item	Quantity	References	Part value	Package and supplier code	Notes
27	1	J4	UART I/F	JACK-3_5-35RASMT2BHNTRX	
28	1	J5	CH2	2 way connector WAGO-250-402	
29	1	J6	CH0	2 way connector WAGO-250-402	
30	1	J7	CH3	2 way connector WAGO-250-402	
31	1	J8	CH1	2 way connector WAGO-250-402	
32	1	J9	2 way connector	2 way connector WAGO-236-402	
33	1	J11	Dimming 0 -10 V	2 pin-pitch 2.54	
34	2	L1,L7	WE-CBF	Capc-0603	
35	4	L2,L3,L4,L5	470 µH	DS5022P-Coilcraft	1,2 A sat
36	1	L6	8,5 µH	ELL4GG Panasonic	
37	4	L8,L9,L10,L11	WE-CBF	Capc-0603	
38	1	P1	RESET	EVQPSM Panasonic	
39	1	P2	LED CH SEL	Smd-switch-B3FS	
40	1	P3	INC	Smd-switch-B3FS	
41	1	P4	DEC	Smd-switch-B3FS	
42	4	Q1,Q2,Q5,Q6	STN3NF06L	SOT223	
43	4	Q3,Q4,Q7,Q8	2N7002	SOT23	
44	10	R1,R2,R3,R9,R77,R34, R35,R56, R57,R68,R78	0R	Resc-0603	
45	1	R4	15 K	Resc-0603	
46	1	R5	4K7	Resc-0603	
47	2	R6,R7	1 K	Resc-0603	
48	1	R8	51.1 K	Resc-0603	
49	1	R10	3.3 K	Resc-0603	
50	1	R11	47 K	Resc-0603	
51	2	R12,R73	100 K	Resc-0603	
52	1	R13	27 K	Resc-0603	

Table 7. Bill of material (continued)

Item	Quantity	References	Part value	Package and supplier code	Notes
53	8	R14,R15,R16,R17,R32, R33, R54,R55	N.M.	Resc-0603	Not mounted
54	12	R18,R19,R24,R25,R40, R41,R46,R47,R67,R69, R72,R75	26K1	Resc-0603	
55	4	R20,R21,R42,R43	10R	Resc-0603	
56	4	R22,R23,R44,R45	160R	Resc-0603	
57	8	R26,R27,R28,R29,R48, R49,R50,R51	1R8	R2512	1 W 1%
58	4	R30,R31,R52,R53	N.M.	Resc-0805	Not mounted
59	6	R36,R37,R58,R59,R70, R76	1K2	Resc-0603	
60	4	R38,R39,R60,R61	5K1	Resc-0603	
61	1	R62	10R	R700DIAM100	1 W
62	1	R63	18 K	Resc-0805	
63	2	R64,R66	10 K	Resc-0805	
64	1	R65	2K7	Resc-0805	
65	2	R71,R74	100 K	Resc-0603	Not mounted
66	1	U1	STMLUX38	TSSOP38	
67	2	U2,U3	PM8834	SOI8	
68	5	U4,U5,U6,U7,U10	LMV321ILT	SOT23-5L	Not mounted
69	1	U8	ST1S14	SO8-EP-9	
70	1	U9	LK112M33TR	SOT23-5L	
71	1	X1	16 MHz-N.M.	X32-3_2x2_5-MEC	Not mounted

12 Revision history

Table 8. Document revision history

Date	Revision	Changes
12-Jun-2013	1	Initial release.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

ST PRODUCTS ARE NOT AUTHORIZED FOR USE IN WEAPONS. NOR ARE ST PRODUCTS DESIGNED OR AUTHORIZED FOR USE IN: (A) SAFETY CRITICAL APPLICATIONS SUCH AS LIFE SUPPORTING, ACTIVE IMPLANTED DEVICES OR SYSTEMS WITH PRODUCT FUNCTIONAL SAFETY REQUIREMENTS; (B) AERONAUTIC APPLICATIONS; (C) AUTOMOTIVE APPLICATIONS OR ENVIRONMENTS, AND/OR (D) AEROSPACE APPLICATIONS OR ENVIRONMENTS. WHERE ST PRODUCTS ARE NOT DESIGNED FOR SUCH USE, THE PURCHASER SHALL USE PRODUCTS AT PURCHASER’S SOLE RISK, EVEN IF ST HAS BEEN INFORMED IN WRITING OF SUCH USAGE, UNLESS A PRODUCT IS EXPRESSLY DESIGNATED BY ST AS BEING INTENDED FOR “AUTOMOTIVE, AUTOMOTIVE SAFETY OR MEDICAL” INDUSTRY DOMAINS ACCORDING TO ST PRODUCT DESIGN SPECIFICATIONS. PRODUCTS FORMALLY ESCC, QML OR JAN QUALIFIED ARE DEEMED SUITABLE FOR USE IN AEROSPACE BY THE CORRESPONDING GOVERNMENTAL AGENCY.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.
Information in this document supersedes and replaces all information previously supplied.
The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2013 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com