Pavlo HERASYMENKO

Institute of Electrodynamics of the National Academy of Sciences of Ukraine
ORCID: 0000-0001-6244-1133

doi:10.15199/48.2023.08.20

Software implementation of pulse-density modulation control for H-bridge series-resonant converters

***Abstract**. The paper presents an approach to the software implementation of pulse-density modulation (PDM) control for H-bridge series-resonant converters. The formation of pulse-density modulated control signals directly by a microcontroller makes it possible to avoid using many additional chips for the hardware implementation of PDM control. For this purpose, four control patterns and transactions between them are described. Moreover, the features of the organization PDM sequences in the source code of the microcontroller are discussed, and an algorithm for choosing control patterns and preparing timer registers is shown.*

Streszczenie.** W artykule przedstawiono podejście do implementacji programowej sterowania modulacją gęstości impulsów (PDM) dla przekształtników szeregowo-rezonansowych z mostkiem H. Tworzenie sygnałów sterujących o modulowanej gęstości impulsów bezpośrednio przez mikrokontroler umożliwia uniknięcie stosowania wielu dodatkowych układów scalonych do sprzętowej implementacji sterowania PDM. W tym celu opisano cztery wzorce sterowania oraz transakcje między nimi. Ponadto omówiono cechy organizacji sekwencji PDM w kodzie źródłowym mikrokontrolera oraz przedstawiono algorytm wyboru wzorców sterowania i przygotowania rejestrów czasowych.* (Implementacja programowa sterowania modulacją gęstości impulsów dla szeregowo-rezonansowych przetwornic z mostkiem H***)

Keywords: control patterns, converter, hardware-software implementation, inverter, pulse-density modulation.
Słowa kluczowe: wzorce sterowania, konwerter, implementacja sprzętowo-programowa, falownik, modulacja gęstości impulsów.

Introduction

Control methods play a very important role in voltage-source converters based on series-resonant inverters, which are widely used in devices for various purposes. In recent years, pulse-density modulation (PDM) control has been most commonly applied because of the ability to achieve soft-switching inverter operation (zero-voltage switching and quasi-zero current switching). However, PDM is more difficult to implement compared with other control methods such as pulse-frequency modulation (PFM), pulse-width modulation (PWM), phase-shift control (PSC), etc.

The core of modern control systems of transistor converters is usually a microcontroller (MCU) [1-4] or field-programmable gate array (FPGA) [5-7]. Both contemporary MCs (including digital signal processors) and FPGAs make it possible to create complex control systems with desired control methods and algorithms. Some authors prefer FPGAs due to their parallel operations [8, 9] (e.g. they can handle parallel inputs) or much more customizable architecture [10], while other authors prefer MCUs due to their much less programming complexity and good cost-effectiveness [11, 12].

The implementation of the PDM control method is more complex compared to control methods such as PWM, PSC, asymmetrical voltage cancellation, and, especially compared to PFM and asymmetrical duty cycle, where only two complementary signals are necessary. A common approach to PDM implementation is the generation of high-frequency control signals and one or two PDM envelope signals, which are applied to an AND-OR logic circuit, with the help of which the resulting control signals are formed [13, 14]. In the case of using an FPGA, all this can be implemented inside a chip [15], and in the case of using an MCU, as a rule, it is implemented externally [16, 17]. For modular converters with interleaved or stepped PDM controls [18] or extended PDM inverters [19], the use of an external logic block will require a large number of additional chips, which reduces the attractiveness of using MCUs.

To get the benefits of FPGA and MCU, as well as mitigate their drawbacks, both can be used in the same control system [20], but this approach is too expensive.

In this paper, the aim is to show software implementation of pulse-density control for H-bridge series-resonant converters. The paper is based on the results obtained in the implementation of PDM control in control systems from earlier publications [18], and [21-23]. Key contributions of this work include the following:

1) The presented software implementation of the traditional PDM control method requires only four control patterns described in the paper, which must be stored in the MCU.

2) The state diagram is shown for the proper transition between the control patterns, allowing the converter to operate with zero-voltage switching for any PDM sequence.

3) Such an approach in software implementation makes it possible to avoid using many additional chips for a hardware implementation of PDM control.

Control system of H-bridge series-resonant converter

A. System description

Fig. 1 shows a simplified circuit configuration of an H-bridge series-resonant converter. This configuration is the conventional H-bridge voltage-source inverter configuration. The DC input voltage (V_d) of the inverter can be supplied from a bridge rectifier, an active power corrector, or another part of the converter, depending on its overall configuration. The series-connected resistor R, inductor L, and capacitor C form a series resonant circuit and are equivalent load parameters.

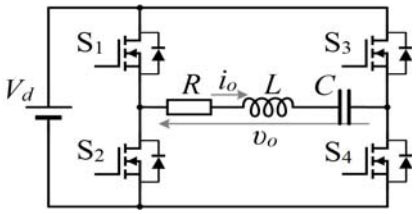


Fig. 1. Conventional H-bridge voltage-source inverter

B. Pulse-density modulation

Fig. 2 shows the PDM patterns of the traditional irregular PDM for a constant value (k-value of 10) of the number of cycles of the output voltage period T_O during the duration T_M of PDM sequences; where m is the number of cycles of T_O within the duration T_M when the inverter acts as a square-wave ac-voltage source ($v_O = \pm V_d$), and n is the number of cycles of T_O within the duration T_M when the inverter acts as a zero-voltage source ($v_O = 0$).

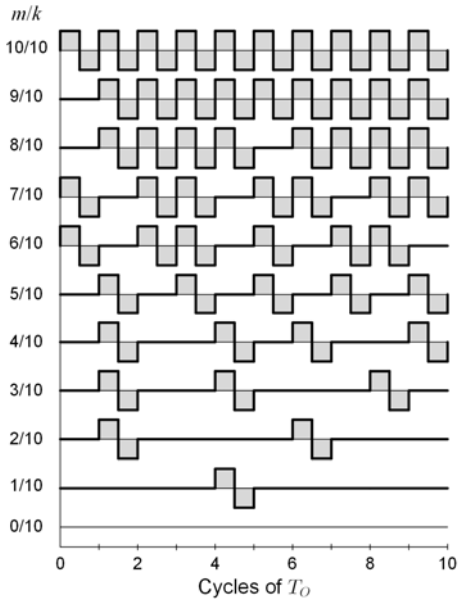


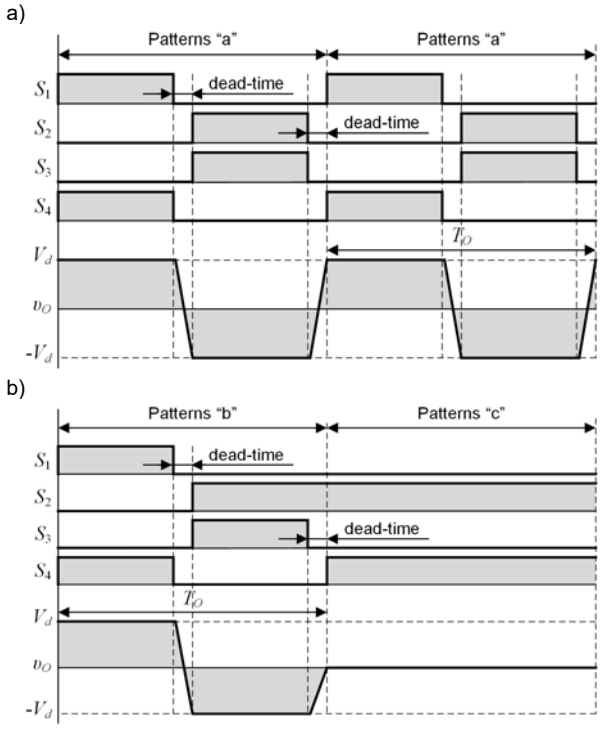
Fig. 2. PDM patterns of the traditional irregular PDM at k = 10

PDM sequences can be represented by “1” and “0” sequences, which is convenient for describing them in the MCU source code; “1” means the inverter acts as a square-wave ac-voltage source, and “0” means the inverter acts as a zero-voltage source.

Control patterns

To avoid overlapping of transistors owing to recharging of their stray capacitances, it is necessary to provide dead time between control signals (S_1-S_4). For this reason, the control patterns of these signals must be different depending on the change of “1” and “0” in a PDM sequence.

Any PDM sequence with the desired PDM patterns can be formed with four control patterns for the following cases: “1” becomes “1”, “1” becomes “0”, “0” becomes “0”, and “0” becomes “1”. Thus, an MCU must implement the four patterns “a”, “b”, “c”, and “d”, which are shown in Fig. 3.



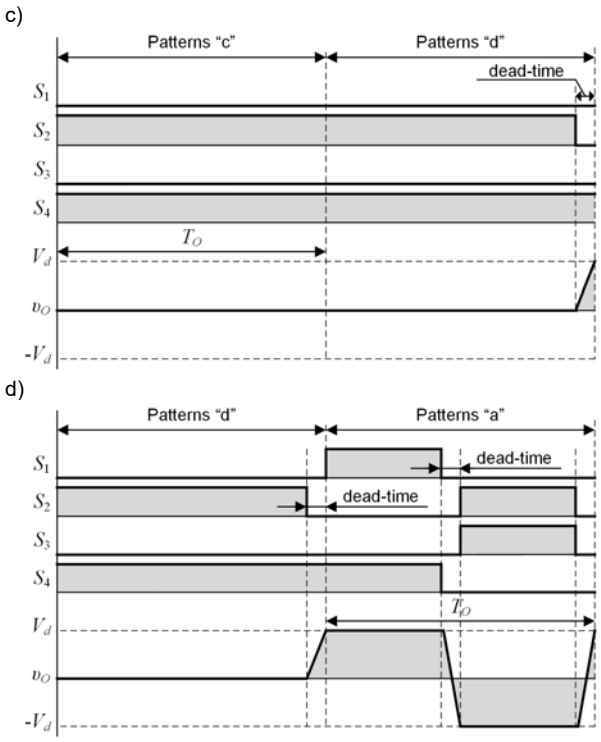


Fig. 3. Control patterns: (a) “1” becomes “1”, (b) “1” becomes “0”, (c) “0” becomes “0”, and (d) “0” becomes “1”

To implement the desired control patterns for the following T_O, they must be prepared (set to the desired values in the MCU timer registers responsible for generating control signals) at the current T_O. At the same time, there is a strict requirement to change the control patterns – the patterns being formed depend on the patterns that follow them.

So, for example, for the sequence “101” in the first T_O (the first “1” in the sequence – Patterns “a” or “b”), the MCU must prepare Patterns “d” (not Patterns “c”) for the next T_O (“0” in the sequence), because “0” is followed by “1”. A state diagram showing the correct change in control patterns depending on the “1/0” sequences is shown in Fig. 4.

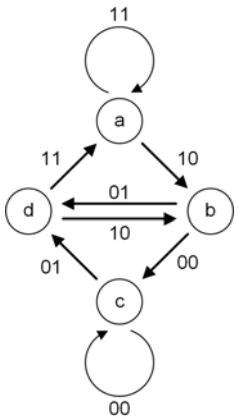


Fig. 4. State diagram for changes between control patterns

Fig. 5 shows an example of the relationship between choosing the control patterns from the “0/1” sequence in accordance with the state diagram in Fig. 4. Thus, it is impossible to get direct transitions between patterns “a” and “c”, transitions from patterns "d" to "c" and from "b" to "a", which would lead to the overlap of transistors. In the meantime, the transitions shown in the stage diagram make it possible to avoid transistors' overlap.

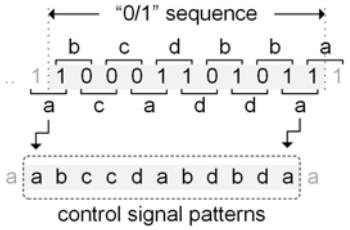


Fig. 5. Example of the relationship between choosing the control patterns from the "0/1" sequence

Software implementation

For software implementation of PDM control, all PDM sequences have to be presented as "1/0" sequences. Fig. 6(a) shows the "1/0" sequences corresponding to the PDM sequences in Fig. 2.

a)

Cycle no.	1	2	3	4	5	6	7	8	9	10
10/10	1	1	1	1	1	1	1	1	1	1
9/10	0	1	1	1	1	1	1	1	1	1
8/10	0	1	1	1	1	0	1	1	1	1
7/10	1	0	1	1	0	1	1	0	1	1
6/10	1	0	1	1	0	1	0	1	1	0
5/10	0	1	0	1	0	1	0	1	0	1
4/10	0	1	0	0	1	0	1	0	0	1
3/10	0	1	0	0	1	0	0	0	1	0
2/10	0	1	0	0	0	0	1	0	0	0
1/10	0	0	0	0	1	0	0	0	0	0
0/10	0	0	0	0	0	0	0	0	0	0

b)

Cycle no.	1	2	3	4	5	6	7	8	9	10	11
10/10	1	1	1	1	1	1	1	1	1	1	1
9/10	1	1	1	1	1	1	1	1	1	0	1
8/10	1	1	1	1	0	1	1	1	1	0	1
7/10	1	0	1	1	0	1	1	0	1	1	1
6/10	1	0	1	1	0	1	0	1	1	0	1
5/10	1	0	1	0	1	0	1	0	1	0	1
4/10	1	0	0	1	0	1	0	0	1	0	1
3/10	1	0	0	1	0	0	0	1	0	0	1
2/10	1	0	0	0	0	1	0	0	0	0	1
1/10	1	0	0	0	0	0	0	0	0	0	1

Fig. 6. "1/0" sequences: (a) corresponding to the PDM sequences in Fig. 2; (b) aligned, start with "1"

The PDM sequence (m;k) = (0;10) can be excluded from the sequence list since it can be considered as turning off the inverter.

In order to choose the correct control patterns at the end of a sequence, it is necessary to know where the next sequence starts – will it be a "1" or a "0"? For this reason, a decision on the next sequence must be made at least one step before the end of the current sequence. On the other hand, it is possible to order the PDM sequences so that their patterns always begin with "1". In such a way there is no need to know which PDM sequence will be the next, as it still starts from "1". This allows making a decision about choosing the next sequence at the last iteration of the sequence, which essentially is more correct.

Fig. 6(b) shows the ordered PDM sequences of Fig.6(a), so all sequences start with "1". To simplify the selection of control patterns, they should be increased by one position, repeating the first position. The PDM sequence (m;k) = (0;10) was excluded.

In the case of using inconstant PDM control (k-value varies depending on PDM sequences) [21], an additional array with k-values is required; in the case of using PS-PDM control [22], only PDM sequences in the pulse-density range for 0.5 to 1 are used, and the same control signal paters are needed.

Within the acting T_O, the MCU must prepare timer registers for the next T_O. To do this, an interrupt can be generated where all calculations will be made and the values of the timer register will be updated.

The algorithm of the MCU program inside this interrupt is as follows.

Algorithm 1. Choosing control patterns and preparing timer registers

```
switch( "1/0" sequences [active seq.][counter] )
    1 | if ( "1/0" sequences [active seq.][counter + 1] ) = 1
      |         Patterns ← "a"
      | else
      |         Patterns ← "b"
    0 | if ( "1/0" sequences [active seq.][counter + 1] ) = 1
      |         Patterns ← "d"
      | else
      |         Patterns ← "c"
Timer registers ← Values ( Patterns, T_O )
counter ← counter + 1
if ( counter = max value [active seq.] )
    | counter ← 0
    | active seq. ← new seq.
```

The "counter" in the algorithm is a variable for determining the cycle number of T_O.

Hardware implementation with MCU

For the software implementation of the previously described patterns, there are certain requirements for the MCU. The timer of the MCU must be able to generate 4 output signals with the ability to set the duration and start of the pulse separately for each signal. Or it should be possible to generate these signals by several synchronized timers of the MCU. Another requirement is a high clocking frequency of the timer/timers, which is caused by a strong change in the phase-shift between current and voltage at the output of the inverter as a result of a change in the duration of the period [23].

Based on previous results obtained with the implementation of PDM control in control systems from earlier publications [18, 21-23], one of the suitable MCUs for this task is the STM32 microcontrollers of the STM32F3, STM32G4, and STM32H7 microcontroller families, which have a high-resolution timer [24].

As an example of the hardware, Fig. 7. shows a block diagram of the control system for implementing PDM control based on the STM32G474 MCU from [21].

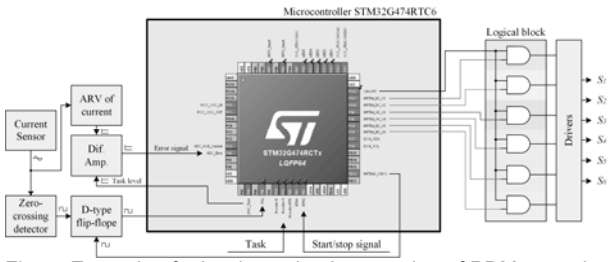


Fig. 7. Example of a hardware implementation of PDM control

In the general case, for the hardware implementation of PDM control, only two microcircuits are required – an MCU and a logic chip of the "AND" gates. However, a phase-locked loop system, which can also be implemented with the same MCU, requires additional components such as a D-type flip-flop [23]. Several op-amps are also needed to regulate the current or power of the inverter (feedback circuit).

The logic chip “AND” if desired, can be excluded. But it allows for simplifying the source code of the MCU and makes it possible immediately switch off all inverter transistors when necessary. Whereas in the case of using the sequence of (0;10) for switching off the transistors, it is necessary to wait for the end of the acting sequence.

Conclusions

This paper proposes a simple yet reliable method for software implementation of PDM control with an MCU for an H-bridge converter. It requires an MCU and a logic chip “AND” for a hardware implementation, allowing to avoid the involvement of additional chips compared to the commonly used approach. For software implementation of PDM control, four control patterns must be stored in the source code of the MCU and organized transactions between them in the way described above. Based on the described feathers of the software implementation, the presented control signals patterns can be easily modified for enhanced PDM controls. In the case of the extended PDM-inverter, more control patterns are required, but the main principles of software implementation are the same.

The main advantage of this software implementation of PDM control with an MCU is its simplicity compared to using an FPGA; its disadvantage is the need for many timer channels in the case of modular PDM converters with interleaved or stepped PDM control methods.

***Authors**: Ph.D. Pavlo Herasymenko (IEEE M’15–SM’22), a Senior Researcher, Department of Transistor Converters, Institute of Electrodynamics of NASU, 56 Beresteysky Avenue, office 457, 03057, Kyiv, Ukraine, E-mail: herasymenko@ieee.org*.

REFERENCES

[1] Kim J.-W., Lee M., and Lai J.-S., A new control method for series resonant inverter with inherently phase-locked coil current with induction cookware applications, *Proc. 2018 IEEE APEC*, 2018, 3517-3522, doi: 10.1109/APEC.2018.8341611

[2] Bayindir N.S., Kukrer O., and Yakup M., DSP-based PLL-controlled 50-100 kHz 20 kW high-frequency induction heating system for surface hardening and welding applications, *IEE Proc. Electric Power Appl.*, 150 (2003), No. 3, 365-371, doi: 10.1049/ip-epa:20030096

[3] Namadmalan A., Universal Tuning System for Series-Resonant Induction Heating Applications, *IEEE Transactions on Industrial Electronics*, 64 (2017), No. 4, 2801-2808, doi: 10.1109/TIE.2016.2638399

[4] Wu K.Y., Zhang T., He Z.W., Jiang X.M., and Liu X.G., STM32-based digital charging power supply, Proc. Int. Conf. Electron., Elect. Eng. and Inf. Science, (2015), 37-47, doi: 10.1142/9789814740135_0005

[5] Jimenez O., Lucia O., Urriza I., Barragan L.A., and Navarro D., Design and Evaluation of a Low-Cost High-Performance Σ–Δ ADC for Embedded Control Systems in Induction Heating Appliances, *IEEE Transactions on Industrial Electronics*, 61 (2002), No. 5, 2601-2611, doi: 10.1109/TIE.2013.2278524

[6] Hassaine L. and Bengourina M.R., Design and digital implementation of power control strategy for grid connected photovoltaic inverter, *International Journal of Power Electronics and Drive System*, 10 (2019), No. 3, 1564-1574, doi: 10.11591/ijpeds.v10.i3.1564-1574

[7] Puyal D., Barragan L.A., Acero J., Burdio J.M., and Millan I., An FPGA-Based Digital Modulator for Full- or Half-Bridge Inverter Control, *IEEE Trans. Power Electron.*, 21 (2006), No. 5, 1479-1483, doi: 10.1109/TPEL.2006.880234

[8] de Castro A., Zumel P., Garcia O., Riesgo T., and Uceda J., Concurrent and simple digital controller of an AC/DC converter with power factor correction based on an FPGA, *IEEE Transactions on Power Electronics*, 18 (2003), No. 1, 334-343, doi: 10.1109/TPEL.2002.807106

[9] Fujimori T. and Watanabe M., Gate density advantage of parallel-operation-oriented FPGA architecture, *Proc. IEEE NAECON*, 2017, 155-158, doi: 10.1109/NAECON.2017.8268761

[10] Jimenez O., Lucia O., Urriza I., Barragan L.A., Mattavelli P., and Boroyevich D., An FPGA-Based Gain-Scheduled Controller for Resonant Converters Applied to Induction Cooktops, *IEEE Transactions on Power Electronics*, 29 (2014), No. 4, 2143-2152, doi: 10.1109/TPEL.2013.2276041

[11] Rajaraman J., Prajapati V., Bhagat M., Rao Y.S., and Sawant R.R., Implementation of a digital controller using DSP TMS320F28335 for frequency and power tracking of load resonant inverters, *Proc. IEEE ICIT*, 2016, 1085-1090, doi: 10.1109/ICIT.2016.7474905

[12] Martin-Segura G., Sala-Perez P., Ferrater-Simon C., Lopez-Mestre J., Bergas-Jane J., and Montesinos-Miracle D., All-digital DSP-based phase-locked loop for induction heating applications, *Int. Trans. Electr. Energ. Syst.*, 23 (2013), No. 7, 1095-1106, doi: 10.1002/etep.1640

[13] Fujita H. and Akagi H., Control and performance of a pulse-density-modulated series-resonant inverter for corona discharge processes, *IEEE Transactions on Industry Applications*, 35 (1999), No. 3, 621-627, doi: 10.1109/28.767013

[14] Sandali A., Cheriti A., and Sicard P., Simple PDM pattern generation for an AC/AC resonant converter, *Proc. IEEE IECON*, 2002, 294-299, doi: 10.1109/IECON.2002.1187524

[15] Esteve V., et al., Enhanced Pulse-Density-Modulated Power Control for High-Frequency Induction Heating Inverters, *IEEE Transactions on Industrial Electronics*, 62 (2015), No. 11, 6905-6914, doi: 10.1109/TIE.2015.2436352

[16] Chen S., Li H., and Tang Y., A burst mode pulse density modulation scheme for inductive power transfer systems without communication modules, *Proc. IEEE APEC*, 2018, 1071-1075, doi: 10.1109/APEC.2018.8341148

[17] Karafil A., Ozbay H., and Oncu S., Design and Analysis of Single-Phase Grid-Tied Inverter With PDM MPPT-Controlled Converter, *IEEE Transactions on Power Electronics*, 35 (2020), No. 5, 4756-4766, doi: 10.1109/TPEL.2019.2944617

[18] Herasymenko P., Pavlovskyi V., Soft start-up strategy of pulse-density-modulated series-resonant converter for induction heating application, *International Journal of Power Electronics and Drive System*, 12 (2021), No. 1, 258-272, doi: 10.11591/ijpeds.v12.i1.pp258-272

[19] Herasymenko P. and Yurchenko O., An Extended Pulse-Density-Modulated Series-Resonant Inverter for Induction Heating Applications, *Proc. IEEE RTUCON*, 2020, pp. 1-8, doi: 10.1109/RTUCON51174.2020.9316617

[20] Sheng X., Shi L., and Fan M., An Improved Pulse Density Modulation of High-Frequency Inverter in ICPT System, IEEE Transactions on Industrial Electronics, 68 (2021), No. 9, 8017-8027, doi: 10.1109/TIE.2020.3013782

[21] Herasymenko P., Pavlovskyi V., Yurchenko O., A series-resonant inverter with extended topology and pulse-density-modulation control for induction heating applications, *International Journal of Power Electronics and Drive Systems*, 13 (2022), No. 1, 348-367, doi: 10.11591/ijpeds.v13.i1.pp348-367

[22] Herasymenko P., Combined PS-PDM control method for voltage-source series-resonant inverter, *Przeglad Elektrotechniczny*, 97 (2021), No. 5, 40-45, doi: 10.15199/48.2021.05.07

[23] Herasymenko P., Inexpensive, high-performance STM32-based software PLL for series-resonant inverters, *Przeglad Elektrotechniczny*, 98 (2022), No. 8, 132-138, doi: 10.15199/48.2022.08.25

[24] AN4539 - Application note “HRTIM cookbook”. Available online: https://www.st.com/content/ccc/resource/technical/document/application_note/13/d6/48/9d/11/11/4c/08/DM00121475.pdf/files/DM00121475.pdf/jcr:content/translations/en.DM00121475.pdf