

# CRON Expressions

We use CRON expressions setup the timing of our hardware during device configuration using the [Lobaro Maintenance Tool](#) or remotely over the air.

The CRON expressions consists of 6 fields, separated by space: e.g. `0 0/15 * * * *`, which describes time points every 15 minutes starting from minute 0.

CRON expression allow you define specific POINTS in time they are NOT helpful to define time durations. They can be seen has the start time to trigger an action, for example initiating the sensor measurement or wireless sendout. The duration of an action is defined by a separate configuration parameter if needed.

## Meaning of the six fields:

`0 0/15 * * * *`

	Second	Minute	Hour	Day of Month	Month of year	Day of Week
Range	(0-59)	(0-59)	(0-23)	(1-31)	(1-12)	(1-7)
Alternative					JAN-DEC	MON-SUN
Allowed special chars	, - * / H L	, - * / H L	, - * / H L	, - * / ? H L W	, - * / H L	, - * / ? H L
Example						1 = MON



H L W are only supported on recent firmware releases, see below for details.



Day of Month and Day of Week should be treated mutually exclusive and one of them should be set to "?"

- `'0 0 1 1 * MON'` Will trigger on 1st of every month **that is a** Monday

- Meaning of the six fields:
- Examples
  - Examples using Extended Crons
- Special Characters:
  - star, asterisk (\*)
  - question mark (?)
  - dash, minus (-)
  - comma (,)
  - slash (/)
- Extended Cron expressions
  - Weekday (W)
  - Last (L)
  - Hash (H) - consistent pseudo random number
    - Pitfalls with using H
- Online CRON Generators and Tester:
- Further Reading

## Examples

Cron definition	Description	Trigger time point(s) (hh:mm:ss)
<code>0 5 * * * *</code>	Hourly at minute 5, second 0	00:05:00, 01:05:00, 02:05:00...
<code>0 1/10 * * * *</code>	Every 10 minutes starting from minute 1, second 0	00:01:00, 00:11:00, 00:21:00, [...], 01:01:00, [...]
<code>0 0 6 * * *</code>	Daily on hour 6, minute 0, second 0	06:00:00
<code>0 0 13 1,15 * *</code>	Hour 13, minute 0, second 0 on day 1 and 15	13:00:00 at 1 <sup>st</sup> and 15 <sup>th</sup> of month
<code>0 15 9 1-5 * *</code>	Hour 9, minute 15, second 0 on day 1 to 5	09:15:00 at 1 <sup>st</sup> to 5 <sup>th</sup> of month
<code>0 0 6 * * 1</code>	Weekly on Monday at 06:00:00	Monday, 06:00:00

## Examples using Extended Crons

These will only work on recent firmware versions, e.g. the [wMBus Gateways](#) versions 0.15.4 and newer.

Cron definition	Description
<code>0 0 11 1W, 15W * *</code>	11am on the days closest to the 1st and the 15th day each month that are not Saturdays or Sundays.

0 0 15 1, L * *	3pm on the first and last day of each month.
0 0 1 ? * 2L	1am on the last Tuesday of each month.
H H H(0- 2) * * *	Every day at a fixed pseudo random time between 00:00:00 and 02:59:59. Same time each day for a given device, but different time for different devices.
H H(0-9) /10 * * * *	Every 10 minutes, but at a fixed time, randomly but consistently picked for each device.



#### Note

Some Lobar nodes do not keep the **real time** internally since for many sensor applications it is enough to configure repetition intervals. For example it may be a important configuration parameter that a specific sensor data gets transmitted every 15 minutes, but it does not matter if the send out takes place on [12:00h,12:15h...] or [12:05h, 12:20h...]. Times are **relative** to the random time when the device (re)starts or the batteries are inserted. If needed by your target application Lobar can deliver on request special firmware support for keeping data acquisition intervals based on a real time clock (RTC) which stays in sync with the real time on your wrist.

## Special Characters:

### star, asterisk (\*)

Used to select all values within a field. For example, "\*" in the minute field means "every minute".

### question mark (?)

Useful when you need to specify something in one of the two fields in which the character is allowed, but not the other. For example, if I want my trigger to fire on a particular day of the month (say, the 10<sup>th</sup>), but don't care what day of the week that happens to be, I would put "10" in the day-of-month field, and "?" in the day-of-week field. See the examples below for clarification.

### dash, minus (-)

Used to specify ranges. For example, "10-12" in the hour field means "the hours 10, 11 and 12".

### comma (,)

Used to specify multiple explicit values. For example, "MON,WED,FRI" in the day-of-week field means "the days Monday, Wednesday, and Friday", "1,15" in the day-of-month field means "on the first and 15th of the month".

### slash (/)

Used to specify increments. For example, "0/15" in the seconds field means "0 and then every 15", that is "the seconds 0, 15, 30, and 45". And "5/15" in the seconds field means "the seconds 5, 20, 35, and 50". You can also use "/" for other fields. "1/3" in the day-of-month field means "fire every 3 days starting on the first day of the month".

## Extended Cron expressions



The following features are only supported in recent firmware.

In 2022 we introduced new features to our Cron Expressions that make them more flexible, but they will not work in older firmware. The [Wireless M-Bus Gateway](#) support them in versions 0.15.4 and higher.

### Weekday (W)

Only valid in the day-of-month field. Used as a modifier added after a given day, e.g. "15W". This will execute on the nearest weekday (Monday to Friday) around that day. So "0 0 12 15W \* \*" will execute on 12h UTC of the 15th of a month, if that day is a Monday, Tuesday, Wednesday, Thursday, or Friday. If the 15th is a Saturday, it will execute on 12h UTC on Friday the 14th. If the 15th is a Sunday, it will execute on 12h UTC on Monday the 16th. It will however never jump into another month. If you use "1W", and the first day of the month is a Saturday, it will execute on Monday the 3rd.

Can be used in combination with lists (comma ", "), must be used for each individual value of the list you want to use it on, e.g. "1W,15W".

Cannot be used with ranges (dash, "-") or increments (slash "/").

Can be used in combination is hash ("H"), in the form "HW" or e.g. "H(6-9)W".

Can be used in combination with "L", but only as "LW", to execute on the last weekday of a month.

## Last (L)

Has a different meaning, depending on where you use it.

A simple "L" means "the last of something". This is useful for the day-of-month field, as the last day of a month is dependent on the current month and year. "0 0 12 L ?" will execute on the 31st of January, on the 28th or 29th (for leap years) of February, on the 31st of March, on the 30th of April, and so on. You can also specify offsets from the last value, like this: "0 0 12 L-3 \* ?". This executes on the 28th of January, on the 25th of 26th of February, and so on.

In the day-of-week field "L" simply refers to Sunday. But in combination with a number, it can be used to execute on the last specific weekday of a month, like this: "0 0 12 ? \* 1L" will execute at 12h on the last Monday of every month.

Can be used in combination with "W", as "LW", to execute on the last weekday of a month.

## Hash (H) - consistent pseudo random number

The character "H" can be used in place of a number in any field of the cron. The "H" will be replaced with a consistent random number, that fits into that field. On any given device this number will always be the same for a field, but it will differ for different devices. This can be used as a simple mechanism for load balancing, or to avoid collisions, if you use a lot of devices.

Numbers will be 0-59 for second and minute fields, 0-23 for the hour field, 0-28 for the day-of-month field (it will never activate on 29th, 30th, or 31st), 1-12 for the month field, and 1-7 for the day-of-week field.

When you, for example, set many devices to the cron "0 0 4 \* \* \*", they will all activate at 4 AM UTC in the same second. For LoRaWAN devices, this will lead to a lot of packet collisions and therefore dataloss, if the devices are in the same area. For NB-IoT/LTE-M devices, it will lead to a concentrated workload on the server handling the requests and the system evaluating the data. This can lead to bad system performance during that spike, while the system is idle most of the day.

If instead of "0 0 4 \* \* \*" you set the devices to "H H 4 \* \* \*", the two H characters will be replaced with a per-device-consistent random number in the range of 0-59, for example "31 47 4 \* \* \*" for one and "17 0 4 \* \* \*" for another device. With this, the devices will all activate in the hour between 4 AM and 5 AM UTC and distribute the workload.

You can use "H" to pick a pseudo random number from a specific range by specifying the range in parenthesis like this: "H(4-8)" will be replaced with any of the numbers 4, 5, 6, 7, 8. The range must only use numbers that are valid for the field "H" is used in.

Can be used inside lists (comma ", "), but the generated value could collide with other entries in list (see below).

Cannot be used with ranges (dash, "-").

Can be used with increments (slash, "/"), but only before the slash, as in "H(0-2)/3", not after it.

## Pitfalls with using H

Working with "H" can be confusing and lead to unexpected results. It can be used in many places, but not all of them are useful.

For example writing "0 H/10 \* \* \* \*" will on most devices create a cron that does *not* activate every 10 minutes. It could evaluate to the expression "0 47/10 \* \* \* \*", which will only activate twice each hour, in minutes 47 and 57, but not during minutes 0-46 at all. To have devices activate every 10 minutes at distributed times, use "0 H(0-9)/10 \* \* \* \*" or better yet: "H H(0-9)/10 \* \* \* \*".

Using "H" in a list works, but it might lead to fewer activations as intended. Putting it in a list twice will not produce multiple activations, because both "H" will evaluate to the same number: "0 0 H,H \* \* \*" could turn into "0 0 3,3 \* \* \*", which is the same as "0 0 3 \* \* \*". Using "0 0 H(0-4),H(8-12) \* \* \*" would work to produce two activations per day.

## Online CRON Generators and Tester:

- <https://crontab-generator.org/> - "command" need to be set to some random string. The generated CRON has no "seconds".
- <https://crontab.guru/> - Also without "seconds" field.
- <https://www.freeformatter.com/cron-expression-generator-quartz.html>

## Further Reading

All our CRON expressions are in the same format as the Java Quartz scheduler, without the optional "year" field and a different way of numbering weekdays. A good documentation including examples can be found here:

- <http://www.quartz-scheduler.org/documentation/quartz-2.3.0/tutorials/crontrigger.html>