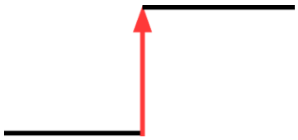
EET165 Lecture #7

1) **Review:** Answer any questions from last week.

2) **Counters:** Counters are a lot like timers. The difference is a timer starts to count based on a time interval as long as the input is 1. A counter counts each time the input gets a pulse that goes from low to high. The input is asynchronous.



3) **Parts of the counters:** Both counters have three variables built into them. Two of the variables are controlled by the programmer and one is controlled by the PLC. There are also 3 outputs that are controlled by the PLC.

   a. The timer has two values that you need to fill in and one that is controlled by the PLC.

      i. **Counter:** Counter is the counter's name, it starts with C5: and then the timer number that can be 0-255. For example: C5:0, C5:8, or C5:100

      ii. **Preset:** The preset is the number the timer will count-up to. The number is between -32768 to 32767 (a signed 16-bit integer).

      iii. **Accum:** Accum is the number that the counter has counted so far.

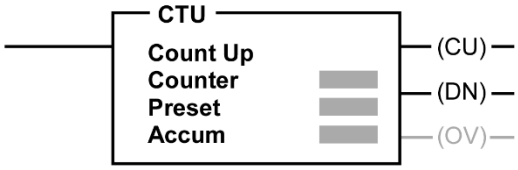
   b. All three timers have three outputs that are controlled by the timers.

      i. **CU / CD:** CU stands for count up and DC stands for count down. This goes high when the input of counter is high.

      ii. **DN:** DN stands for done. DN goes high when the Accum is larger than the Preset. This is true for both the count-up and count-down.

      iii. **OV / UD:** OV stands for overflow. This output goes high when the counter (counting clock pulses) rolls over from 32767 to -32768. UD stands for underflow. This output goes high when the counter (counting clock pulses) rolls over from -32768 to 32767.

4) **CTU:** There are two types of counters, count up and count down. CTU stands for count up.



To use CTU you need to fill in two pieces of information, the counter name, and the preset value. Just like the outputs are in file 0, the inputs are in file 1, bit storage is in file 3, and timers are in file 4 – the counters are in file 5. For example, counter 3, set for a count of 10 would be:

Counter: **C5:3**
Preset: **10**

When the input is 0 or when the input is 1, nothing changes. The Accum is updated only when there is a transition from low to high.

When there is a low to high transition:

ACCUM: Accum is incremented by one each time there is a low to high transition, even if it is greater than the preset (unlike a timer that stops).

CU: CU is always the same as the input.

(*if Accum < Preset then*) **DN:** DN is low, because the Accum is < than the Preset

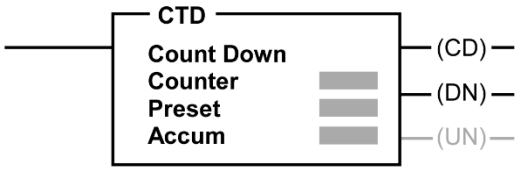
(*if Accum >= Preset then*) **DN:** DN is high, because the Accum is >= to the Preset.

OV: The overflow is zero unless the Accum rolls around from 32767 to -32768.

RES: The only way to reset the counter is with the reset command as shown below.

You can access the outputs using:
C5:3/CU
C5:3/DN
C5:3/OV

5) **CTD:** There are two types of counters, count up and count down. CTD stands for count down.



To use CTD you need to fill in two pieces of information, the counter name, and the preset value. For example, counter 3, set for a count of -5 would be:

Counter: **C5:3**
Preset: **- 5**

When the input is 0 or when the input is 1, nothing changes. The values are updated only when there is a transition from low to high.

When there is a low to high transition:

ACCUM: Accum is incremented by one each time there is a transition, even if it is greater than the preset (unlike a timer that stops).

CD: CD is the count-down bit, and it is always the same as the input.

(*if Accum < Preset then*) **DN:** DN is low, because the Accum is < than the Preset

(*if Accum >= Preset then*) **DN:** DN is high, because the Accum is >= to the Preset.

UN: The underflow is zero unless the Accum rolls around from -32768 to 32767.

RES: The only way to reset the counter is with the reset command as shown below.

You can access the outputs using:
C5:3/CD
C5:3/DN
C5:3/UN

6) **Similar / Different:** Some of the bits work the same and some are the opposite. If you understand how the bits work, it will help you remember how the counter works.

   a. **What is the same:** The DN bit works the same on both counters, that is why both counter types have the same name (DN). When Accum<Preset then DN is 0. When Accum >= Preset then DN is 1.

   b. **What is different:** A CTU's preset is normally positive and when the CTU gets a pulse, Accum is incremented. So, the Accum is moving towards preset.

      A CTD's preset is normally negative and when the CTD gets a pulse, Accum is decremented. So, the Accum is moving towards the preset.

7) **Examples:** When the Accum is larger than or equal to the Preset, DN is high. This is true for both counter types. As the Accum gets a pulse, it should move towards the preset value.

Normally, a CTU has a positive preset. On reset, Accum starts at zero and the Accum is smaller than the positive Preset – so the done (DN) starts at 0. When the CTU gets a pulse, Accum goes up. As the Accum gets incremented, it eventually is greater than or equal to the preset and DN turns on. For example:

Pulse	CTU Preset		CTU Accum	CTU DN
1	4	>	0	0
2	4	>	1	0
3	4	>	2	0
4	4	>	3	0
5	4	=	4	1
6	4	<	5	1
7	4	<	6	1

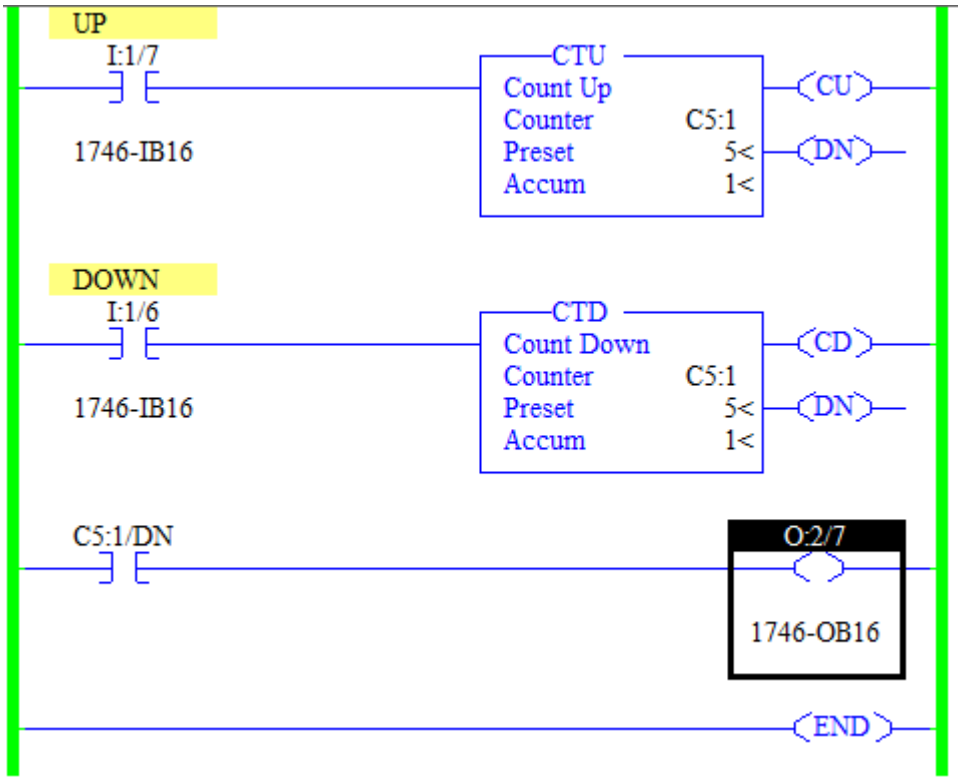
Normally, a CTD has a negative preset. On reset, Accum starts at zero and the Accum is larger than the negative Preset – so the done (DN) starts at 1. When the CTD gets a pulse, Accum goes down. As the Accum gets decremented, it eventually goes below the preset and turns the DN off. For example:

Pulse	CTD Preset		CTD Accum	CTD DN
1	-4	<	0	1
2	-4	<	-1	1
3	-4	<	-2	1
4	-4	<	-3	1
5	-4	=	-4	1
6	-4	>	-5	0
7	-4	>	-6	0

Giving a CTD counter a positive preset will make the Accum get farther and farther away from the preset. DN will only go high if you go so far negative that the number underflows from -32768 to 32767. For example:

Pulse	CTD Preset		CTD Accum	CTD DN
1	6	>	0	0
2	6	>	-1	0
3	6	>	-2	0
4	6	>	-3	0
5	6	>	-4	0
6	6	>	-5	0
7	6	>	-6	0

8) **Up and Down:** A CTU and a CTD can be connected to the same counter in memory. If you do this, you can count up and down. This is useful if you are counting objects, but if there is a rejected part, it can be subtracted from the count.



9) **OV / UN explained:** Two's compliment is a way to store negative numbers in binary. To convert a negative number to its 2's complement representation you must perform 2 steps.

- **Invert all bits (this is called 1's compliment)**
- **Add 1 to the inverted bits to make it 2's compliment**.

It is easy to spot a negative number in binary; the first digit is a 1. A small example is shown below.

Binary Value	Unsigned value	Signed Value
000	**0**	**0**
001	**1**	**1**
010	**2**	**2**
011	**3**	**3**
100	**4**	**-4**
101	**5**	**-3**
110	**6**	**-2**
111	**7**	**-1**

An overflow happens when you add 1_2 to 111_2, this will result in 1000_2. But there is no place for the 1 at the front of the number because this can only hold 3 bits. So, the number becomes 000_2 and when this happens the overflow bit (OV) goes high.

The same thing happens when you subtract 1_2 from 000_2. The result is 111_2 so it wraps the other way. This is called an underflow and when this happens the underflow bit (UN) goes high.

The Accum value is just a larger binary value, it has 16 bits instead of 3 bits. So, it can count from -32768 to 32767.

32767 + 1 = -32768 and the Overflow bit (OV) goes high.

-332768 – 1 = 32767 and the Underflow bit (UN) goes low.