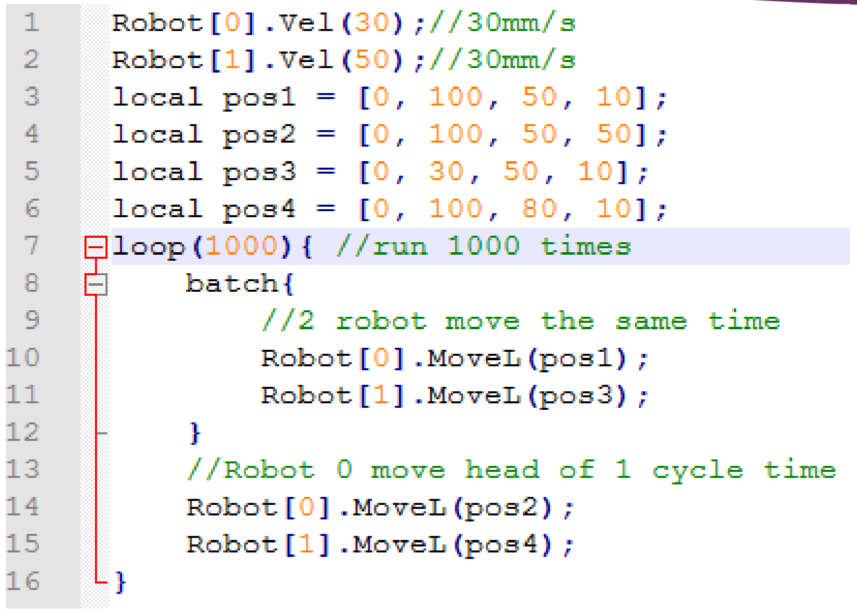
PRC Language

AN EXTENDED C & MATLAB, ACSPL+ FOR ROBOT CONTROLLER

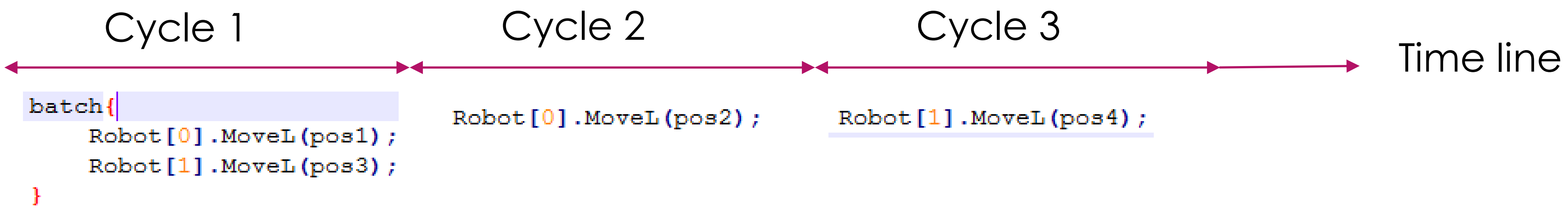
PRESTO SOLUTION, by tuandang@prestosolution.co.kr

Real-time commands



```
Robot[0].Vel(30);//30mm/s
Robot[1].Vel(50);//30mm/s
local pos1 = [0, 100, 50, 10];
local pos2 = [0, 100, 50, 50];
local pos3 = [0, 30, 50, 10];
local pos4 = [0, 100, 80, 10];
loop(1000){ //run 1000 times
    batch{
        //2 robot move the same time
        Robot[0].MoveL(pos1);
        Robot[1].MoveL(pos3);
    }
    //Robot 0 move head of 1 cycle time
    Robot[0].MoveL(pos2);
    Robot[1].MoveL(pos4);
}
```

- Each command will be executed in 1 cycle time.
- Batch of commands also be executed in 1 cycle time.
- Applications:
    - Deterministic behaviors
    - Schedule jobs
    - Create wave form easily

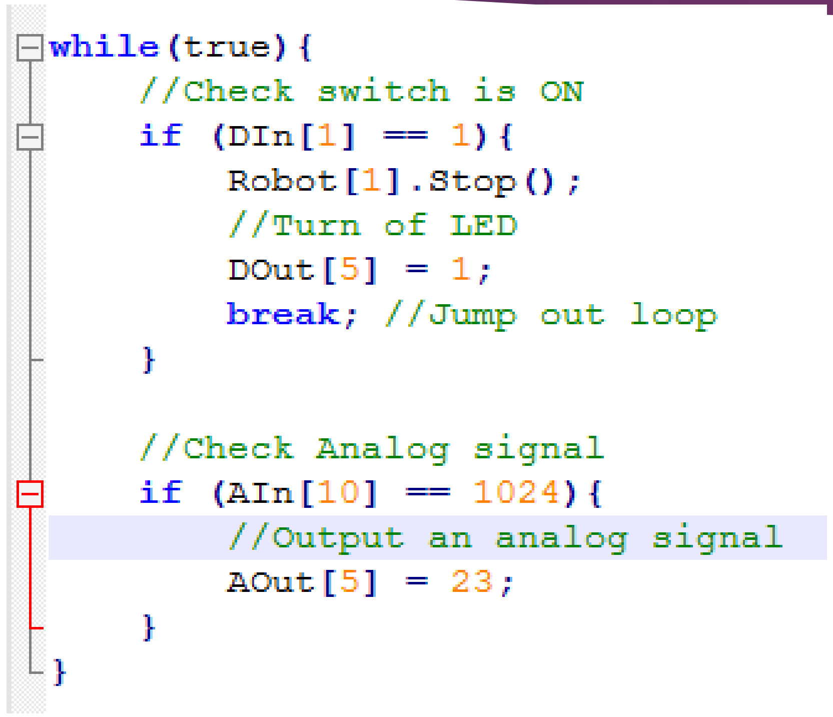


```
batch{
    Robot[0].MoveL(pos1);
    Robot[1].MoveL(pos3);
}
```

```
Robot[0].MoveL(pos2);
```

```
Robot[1].MoveL(pos4);
```

Direct control physical devices



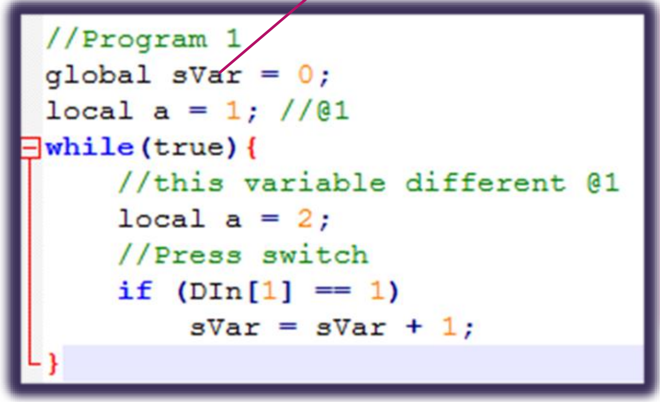
```
while(true){
    //Check switch is ON
    if (DIn[1] == 1){
        Robot[1].Stop();
        //Turn of LED
        DOut[5] = 1;
        break; //Jump out loop
    }

    //Check Analog signal
    if (AIn[10] == 1024){
        //Output an analog signal
        AOut[5] = 23;
    }
}
```

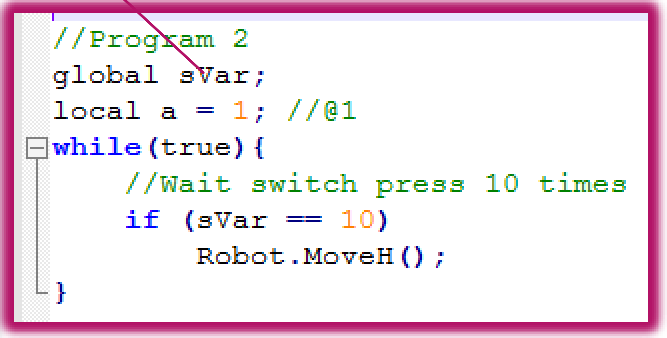
- Control robots and input/output via auto mapping variables easily.
- Multiple robots and input/output support

Global variables cross different programs

sVar: shard between 2 programs



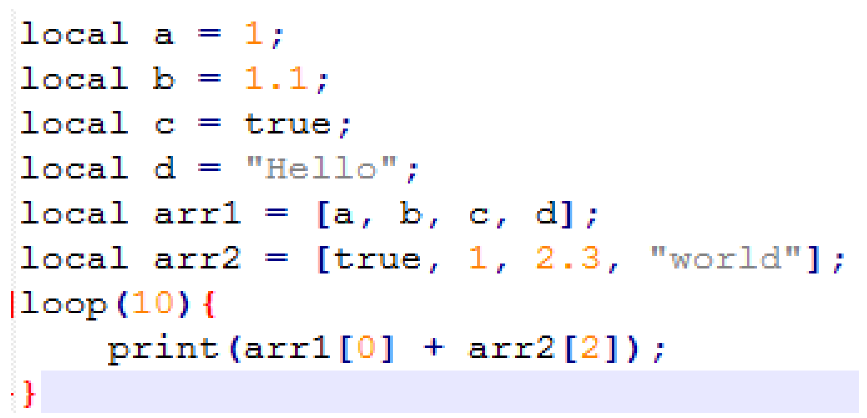
```
//Program 1
global sVar = 0;
local a = 1; //@1
while(true){
    //this variable different @1
    local a = 2;
    //Press switch
    if (DIn[1] == 1)
        sVar = sVar + 1;
}
```



```
//Program 2
global sVar;
local a = 1; //@1
while(true){
    //Wait switch press 10 times
    if (sVar == 10)
        Robot.MoveH();
}
```

- Different programs can shared variables to synchronize with each other.
- Declare variable by keyword global: all variable has the same name, different programs could be shared.

Simple, powerful language



```
local a = 1;
local b = 1.1;
local c = true;
local d = "Hello";
local arr1 = [a, b, c, d];
local arr2 = [true, 1, 2.3, "world"];
loop(10){
    print(arr1[0] + arr2[2]);
}
```

- Dynamic type
- Flexible array for position management
- Simple syntax
- Easy extended for future functions
- Library supports:
  - Motion: Multiple axes, Delta, Puma, Scara
  - Vision