## Three dimension car park

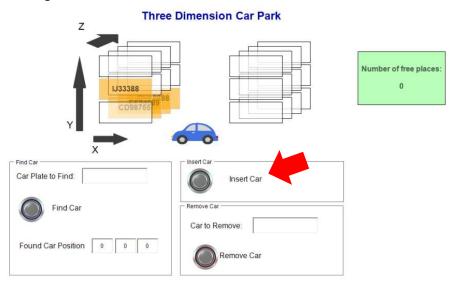
Application to handle cars inside a car park. When a car is inserted into the car park, the PLC program has to first find a free space to ensure where to place the car.

When a car is picked up by the car driver, the car must be located using its car plate number.

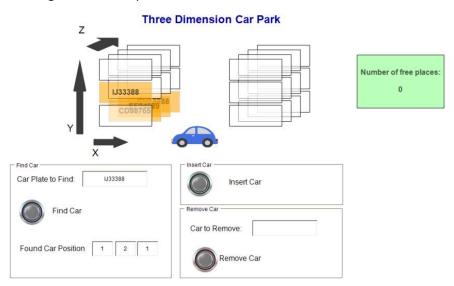
The car plate is used as unique identifier.

It's also possible to check the current position of the car inside the car park. The result of the car position check are x,y,z coordinates of place location in car park.

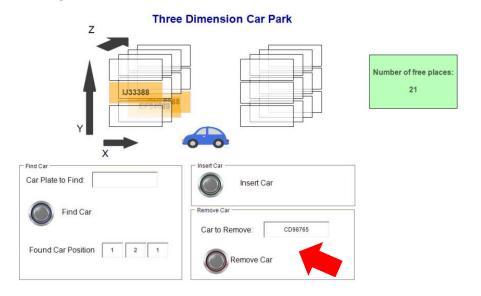
## Inserting cars:

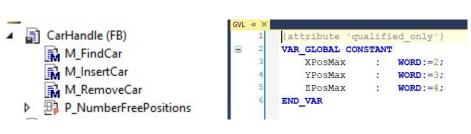


# Searching for car in car park



## Removing a car:





```
FUNCTION_BLOCK CarHandle
    VAR_INPUT
    END VAR
    VAR OUTPUT
    END VAR
    VAR IN OUT
                             ARRAY[*,*,*] OF STRING(15); // pointer to 3d array
        ArrayPark
    END_VAR
    VAR
        Loop
                             CarPos;
                                                          // temp. variable
                                                          // total number of car positions in car park
11
        TotalPositions :
                             INT;
    END_VAR
13
                                                                                                               130
    This FB handles three dimensional car park.
    - ArrayPark - Dimension of the car park
    // copy ARRAY to local variables
    Loop.X := DINT_TO_INT(UPPER_BOUND(ArrayPark, 1));
                                                               // cordinate X
                                                              // cordinate Y
// cordinate Z
    Loop.Y := DINT_TO_INT(UPPER_BOUND(ArrayPark, 2));
    Loop.Z := DINT TO INT (UPPER BOUND (ArrayPark, 3));
11
12
    // total number of car positions in car park
    TotalPositions := Loop.X * Loop.Y * Loop.Z;
```

```
METHOD M InsertCar : BOOL
        VAR INPUT
                            : STRING(15);
                                                 // plate number of the car
            CarPlate
        END VAR
-
        VAR
            CarPos
                            : CarPos;
                                                 // location in car park
        END VAR
                                                                                                                      130
        This Method inserts one car into car park.
        CarPlate - car number plate which should be insert into car park
        Methods Returns TRUE if car was successfully inserted, otherwise FALSE
        // Loop over all position in the 3D car park
        FOR CarPos.X:=1 TO Loop.X DO
=
            FOR CarPos.Y:=1 TO Loop.Y DO
FOR CarPos.Z:=1 TO Loop.Z DO
// insert car at the first free location in the car park
   13
                     IF ArrayPark[CarPos.X, CarPos.Y, CarPos.Z]=' THEN
   14
                         ArrayPark[CarPos.X, CarPos.Y, CarPos.Z] := CarPlate;
   15
                         M InsertCar := TRUE;
   16
                         RETURN;
   17
                    END IF
   18
                END_FOR
   19
   20
            END FOR
   21
        END FOR
        M_InsertCar := FALSE;
        METHOD M FindCar : BOOL
        VAR_INPUT
                            : STRING(15);
                                                  // plate number of the car
            CarPlate
        END VAR
        VAR
            CarPos
                                 CarPos;
                                                  // location in car park
        END VAR
        VAR OUTPUT
            FoundCarPos
                            : CarPos;
                                                  // location in car park
        END VAR
                                                                                                                      130
        This Method finds car in car park using its car plate.
        CarPlate - car number plate which should be looked for in car park
        Methods Returns TRUE if car was successfully found, otherwise FALSE
        // Loop over all position in the 3D car park
        FOR CarPos.X:=1 TO Loop.X DO
            FOR CarPos.Y:=1 TO Loop.Y DO
   10
                FOR CarPos.Z:=1 TO Loop.Z DO
11
                     \label{eq:carpos.X}  \text{IF} \quad \texttt{FIND} (\texttt{ArrayPark}[\texttt{CarPos.X}, \texttt{CarPos.Y}, \texttt{CarPos.Z}], \texttt{CarPlate}) \, > \, 0 \, \, \, \\  \text{THEN} 
   12
   13
                         FoundCarPos := CarPos; // copy coordinates where car was located
   14
                         M FindCar := TRUE;
   15
                         RETURN;
   16
                    END IF
                END FOR
   18
           END FOR
   19
        END FOR
   20
       M_FindCar := FALSE;
                                                                                                                        130
```

```
METHOD M_RemoveCar : BOOL
       VAR_INPUT
                                STRING(15);
                                                // plate number of the car
           CarPlate
       END_VAR
           CarPos
                                CarPos;
                                                 // location in car park
       END_VAR
                                                                                                                     130
       This Method removes one car in car park.
       CarPlate - car number plate which should be removed in car park
       Methods Returns TRUE if car was successfully removed, otherwise FALSE
       // Loop over all position in the 3D car park FOR CarPos.X:=1 TO Loop.X DO \,
FOR CarPos.Y:=1 TO Loop.Y DO
                FOR CarPos.Z:=1 TO Loop.Z DO
   12
                        IF FIND(ArrayPark[CarPos.X, CarPos.Y, CarPos.Z], CarPlate) > 0 THEN
                            ArrayPark[CarPos.X, CarPos.Y, CarPos.Z]:=''; // remove car = set string empty
   14
                            M RemoveCar := TRUE;
   15
                            RETURN;
   16
                        END IF
   17
   18
                END FOR
           END FOR
   19
   20
       END FOR
   21
       M_RemoveCar := TRUE;
                                                                                                                     130
```

## P\_NumberFreePositions



```
PROPERTY P_NumberFreePositions : INT

// Property returns number of free locations in car park
```

```
CarHandle.P_NumberFreePositions.Get 💠 🗴
        VAR
                                                  // number of occupied par places in car park
            CarCount
                                 INT;
                                 CarPos;
                                                  // location in car park
        END VAR
                                                                                                                     140
        // Loop over all position in the 3D car park
        FOR CarPos.X:=1 TO Loop.X DO
    3
            FOR CarPos.Y:=1 TO Loop.Y DO
FOR CarPos.Z:=1 TO Loop.Z DO
6
                    IF ArrayPark[CarPos.X, CarPos.Y, CarPos.Z] <> '' THEN
8
                        CarCount := CarCount+1;
                    END IF
   10
                END_FOR
   11
            END FOR
        END_FOR
   12
   13
   14
        P NumberFreePositions := TotalPositions - CarCount;
```