2025/11/05 02:54 1/1 List Motion

Code	 1
C#	1

2025/11/05 02:54 1/7 List Motion

## **List Motion**

× ListMotion

## Code

```
C#
```

```
private void btnTest_Click(object sender, EventArgs e)
  //
  // 가
                                                                  가 ,
                                                            (
  //
  //
                      (0 \sim 7)
  //
  // 0~31
                                            Mask,
  // 1,2,3
                 axisMask1 = 14
    uint axisMask1 = 0;
  // 32~63
                                             Mask,
    uint axisMask2 = 0;
    int speedMode = (int)ec.EEcmSpeedMode.ecmSMODE TRAPE;
    int stepID = 0;
    if (axisList.Count() < 31)</pre>
        axisMask1 = (uint)(1 << axisID);</pre>
        axisMask2 = 0;
    }
    else
    {
        axisMask1 = 0;
        axisMask2 = (uint)(0x01 << (axisID - 32));
    }
    lmMapIndex = 0;
    double initSpeed = 0;
    double endSpeed = 0;
    double workSpeed = 0;
    double accel = 0;
    double decel = 0;
  //
  //
                                               , ecmLmCtl Run()
    ec.ecmLmCtl_Begin(netID, lmMapIndex, axisMask1, axisMask2, ref
```

2025/11/05 02:54 2/7 List Motion

```
errorCode);
    // lmMapIndex
    ec.ecmLmCtl ClearQue(netID, lmMapIndex, ref errorCode);
  //
                          가
    initSpeed = 0;
    endSpeed = 20000;
    accel = 10000;
    decel = 0; //
                               decel = 0
    workSpeed = endSpeed; //
                                            workSpeed
                                                       endSpeed
    ec.ecmSxCfg_SetSpeedPatt(netID, axisID, speedMode, initSpeed,
endSpeed, workSpeed, accel, decel, ref errorCode);
                 ID
  //
    ec.ecmLmCfg SetStepId(netID, lmMapIndex, stepID++, ref errorCode);
    ec.ecmSxMot MoveStart(netID, axisID, 20000, ref errorCode);
  //
    initSpeed = endSpeed; //
endSpeed가 InitSpeed가
    endSpeed = 40000; //
    accel = 20000;
    decel = 0;
                                           workSpeed endSpeed
    workSpeed = endSpeed; //
    ec.ecmSxCfg_SetSpeedPatt(netID, axisID, speedMode, initSpeed,
endSpeed, workSpeed, accel, decel, ref errorCode);
  //
    ec.ecmLmCfg_SetStepId(netID, lmMapIndex, stepID++, ref errorCode);
    ec.ecmSxMot MoveStart(netID, axisID, 50000, ref errorCode);
                          가
  //
    initSpeed = endSpeed; //
endSpeed가 InitSpeed가
    endSpeed = 40000; //
    accel = 10000:
    decel = 10000;
    workSpeed = 50000;
    ec.ecmSxCfg SetSpeedPatt(netID, axisID, speedMode, initSpeed,
endSpeed, workSpeed, accel, decel, ref errorCode);
  //
    ec.ecmLmCfg SetStepId(netID, lmMapIndex, stepID++, ref errorCode);
    ec.ecmSxMot MoveStart(netID, axisID, 100000, ref errorCode);
  // 가
    initSpeed = endSpeed; //
endSpeed가 InitSpeed가
    endSpeed = 20000; //
    accel = 0; // 가
                               accel = 0
    decel = 20000;
    workSpeed = initSpeed; //
                                                    , workSpeed
```

2025/11/05 02:54 3/7 List Motion

```
, workSpeed = initSpeed 가
    ec.ecmSxCfg SetSpeedPatt(netID, axisID, speedMode, initSpeed,
endSpeed, workSpeed, accel, decel, ref errorCode);
    ec.ecmLmCfg_SetStepId(netID, lmMapIndex, stepID++, ref errorCode);
    ec.ecmSxMot MoveStart(netID, axisID, 70000, ref errorCode);
  //
    initSpeed = endSpeed; //
endSpeed가 InitSpeed가
    endSpeed = 0; //
    accel = 0; // 가
                              accel = 0
    decel = 10000;
    workSpeed = initSpeed;
    ec.ecmSxCfg SetSpeedPatt(netID, axisID, speedMode, initSpeed,
endSpeed, workSpeed, accel, decel, ref errorCode);
  //
    ec.ecmLmCfg_SetStepId(netID, lmMapIndex, stepID, ref errorCode);
    ec.ecmSxMot MoveStart(netID, axisID, 20000, ref errorCode);
    ec.ecmLmCtl_Run(netID, lmMapIndex, ref errorCode);
    int runStepCount = 0, runStepID = 0, runStepState = 0;
    int timeLimit = 100000;
    Stopwatch sw = new Stopwatch();
    sw.Start();
    bool isSuccess = false;
             timeLimit
    Task.Factory.StartNew(() =>
        while (sw.ElapsedMilliseconds < timeLimit)</pre>
        {
      //
            ec.ecmLmSt GetRunStepInfo(netID, lmMapIndex, ref
runStepCount, ref runStepID, ref runStepState, ref errorCode);
            // runStepID :
                                          StepID
            // runStepState :
                                                       (Ready, Busy,
                                             Step
Paused, Completed)
                                                               가
                       StepID가
                                           StepID
Complete
            // StepCount
                          ecmLmSt GetRemStepCount
RemStep
            if (runStepID == stepID && runStepState ==
(int)ec.EEcmLmCmdItemSts.ecmLM CMDITEM STS COMPLETED)
                isSuccess = true;
                break;
```

2025/11/05 02:54 4/7 List Motion

```
}
      //
            // lblRunStepCount.BeginInvoke(new Action(() =>
lblRunStepCount.Text = runStepCount.ToString()));
            // lblRunStepID.BeginInvoke(new Action(() =>
lblRunStepID.Text = runStepID.ToString()));
            // lblRunStepState.BeginInvoke(new Action(() =>
lblRunStepState.Text =
((ec.EEcmLmCmdItemSts)runStepState).ToString()));
            Thread.Sleep(10);
        }
        if (!isSuccess)
      //
        }
    //
        ec.ecmLmCtl End(netID, lmMapIndex, ref errorCode);
    });
    if (!isSuccess)
    //
    }
}
```

```
private void btnTest2_Click(object sender, EventArgs e)
  //
  //
  //
                    (0~7)
  //
        8
    lmMapIndex = 0;
    int ixMapIndex = 0;
  // 0~31
                                          Mask,
  // 1,2,3
               axisMask1 = 14
    uint axisMask1 = 0;
  // 32~63
                                           Mask,
    uint axisMask2 = 0;
    int axisX = axisList[cbxAxisX.SelectedIndex];
    int axisY = axisList[cbxAxisY.SelectedIndex];
    // axisX < 32 \& axisY < 32
    axisMask1 = (uint)((1 << axisX) + (1 << axisY));
    // lmMapIndex
    ec.ecmLmCtl ClearQue(netID, lmMapIndex, ref errorCode);
  //
```

2025/11/05 02:54 5/7 List Motion

```
//
                                            , ecmLmCtl_Run()
    ec.ecmLmCtl Begin(netID, lmMapIndex, axisMask1, axisMask2, ref
errorCode);
                   axisList
  //
    int[] ixAxisList = new int[2]{axisX, axisY};
  //
    ec.ecmIxCfg MapAxes(netID, ixMapIndex, 2, ixAxisList, ref
errorCode);
    int speedType = 1; //VectorSpeed;
    int speedMode = (int)ec.EEcmSpeedMode.ecmSMODE TRAPE;
  //
                       (가
    // endSpeed = workSpeed. decel = 0;
    ec.ecmIxCfg SetSpeedPatt(netID, ixMapIndex, speedType, speedMode,
0, 10000, 10000, 100000, 0, ref errorCode);
    int stepID = 0;
    ec.ecmLmCfg SetStepId(netID, lmMapIndex, stepID++, ref errorCode);
    ec.ecmIxMot LineTo(netID, lmMapIndex, new double[] { 0, 0 }, ref
errorCode);
  //
    // initSpeed, endSpeed = workSpeed. accel, decel = 0;
    ec.ecmIxCfg SetSpeedPatt(netID, ixMapIndex, speedType, speedMode,
0, 10000, 10000, 100000, 0, ref errorCode);
    ec.ecmLmCfg SetStepId(netID, lmMapIndex, stepID++, ref errorCode);
    ec.ecmIxMot LineTo(netID, lmMapIndex, new double[] { 20000, 20000
}, ref errorCode);
    ec.ecmLmCfg SetStepId(netID, lmMapIndex, stepID++, ref errorCode);
    ec.ecmIxMot LineTo(netID, lmMapIndex, new double[] { 50000, 20000
}, ref errorCode);
    ec.ecmLmCfg SetStepId(netID, lmMapIndex, stepID++, ref errorCode);
    ec.ecmIxMot ArcAng A(netID, lmMapIndex, 50000, 30000, 90, ref
errorCode);
    ec.ecmLmCfg SetStepId(netID, lmMapIndex, stepID++, ref errorCode);
    ec.ecmIxMot_LineTo(netID, lmMapIndex, new double[] { 60000, 50000
}, ref errorCode);
    ec.ecmLmCfg SetStepId(netID, lmMapIndex, stepID++, ref errorCode);
    ec.ecmIxMot ArcAng A(netID, lmMapIndex, 50000, 50000, 90, ref
errorCode);
    ec.ecmLmCfg SetStepId(netID, lmMapIndex, stepID++, ref errorCode);
    ec.ecmIxMot_LineTo(netID, lmMapIndex, new double[] { -10000, 60000
```

2025/11/05 02:54 6/7 List Motion

```
}, ref errorCode);
    ec.ecmLmCfg SetStepId(netID, lmMapIndex, stepID++, ref errorCode);
    ec.ecmIxMot ArcAng A(netID, lmMapIndex, -10000, 50000, 90, ref
errorCode);
    ec.ecmLmCfg SetStepId(netID, lmMapIndex, stepID++, ref errorCode);
    ec.ecmIxMot LineTo(netID, lmMapIndex, new double[] { -20000, 30000
}, ref errorCode);
    ec.ecmLmCfg SetStepId(netID, lmMapIndex, stepID++, ref errorCode);
    ec.ecmIxMot ArcAng A(netID, lmMapIndex, -10000, 30000, 90, ref
errorCode):
    ec.ecmLmCfg SetStepId(netID, lmMapIndex, stepID++, ref errorCode);
    ec.ecmIxMot LineTo(netID, lmMapIndex, new double[] { 10000, 20000
}, ref errorCode);
   // initSpeed = workSpeed. accel, endSpeed = 0;
    ec.ecmIxCfg SetSpeedPatt(netID, ixMapIndex, speedType, speedMode,
0, 10000, 10000, 100000, 0, ref errorCode);
    ec.ecmLmCfg_SetStepId(netID, lmMapIndex, stepID, ref errorCode);
    ec.ecmIxMot LineTo(netID, lmMapIndex, new double[] { 0, 0 }, ref
errorCode);
  //
    ec.ecmLmCtl Run(netID, lmMapIndex, ref errorCode);
    int runStepCount = 0, runStepID = 0, runStepState = 0;
    const int timeLimit = 10000;
    Stopwatch sw = new Stopwatch();
    sw.Start();
    bool isSuccess = false;
  //
             timeLimit
  //
              timeLimit
    Task.Factory.StartNew(() =>
    {
        while (sw.ElapsedMilliseconds < timeLimit)</pre>
        {
      //
            ec.ecmLmSt GetRunStepInfo(netID, lmMapIndex, ref
runStepCount, ref runStepID, ref runStepState, ref errorCode);
            // runStepID :
                                           StepID
            // runStepState :
                                              Step
                                                       (Ready, Busy,
Paused, Completed)
      //
                        StepID가
                                            StepID
                                                                가
```

2025/11/05 02:54 7/7 List Motion

```
Complete
            // StepCount
                                    ecmLmSt GetRemStepCount
RemStep
            if (runStepCount == stepID && runStepState ==
(int)ec.EEcmLmCmdItemSts.ecmLM CMDITEM STS COMPLETED)
                isSuccess = true;
                break;
            }
            lblRunStepCount.BeginInvoke(new Action(() =>
lblRunStepCount.Text = runStepCount.ToString()));
            lblRunStepID.BeginInvoke(new Action(() =>
lblRunStepID.Text = runStepID.ToString()));
            lblRunStepState.BeginInvoke(new Action(() =>
lblRunStepState.Text =
((ec.EEcmLmCmdItemSts)runStepState).ToString()));
            Thread.Sleep(10);
        }
        if (!isSuccess)
        {
      //
        }
    //
        ec.ecmLmCtl End(netID, lmMapIndex, ref errorCode);
    });
    if (!isSuccess)
    {
    //
}
```

```
From:
```

http://comizoa.com/info/ - -

Permanent link:

http://comizoa.com/info/doku.php?id=platform:ethercat:100 api:20 eg:listmotion&rev=1574304285

Last update: 2024/07/08 18:22