

..... 1

example 1

C# 1

..... 4

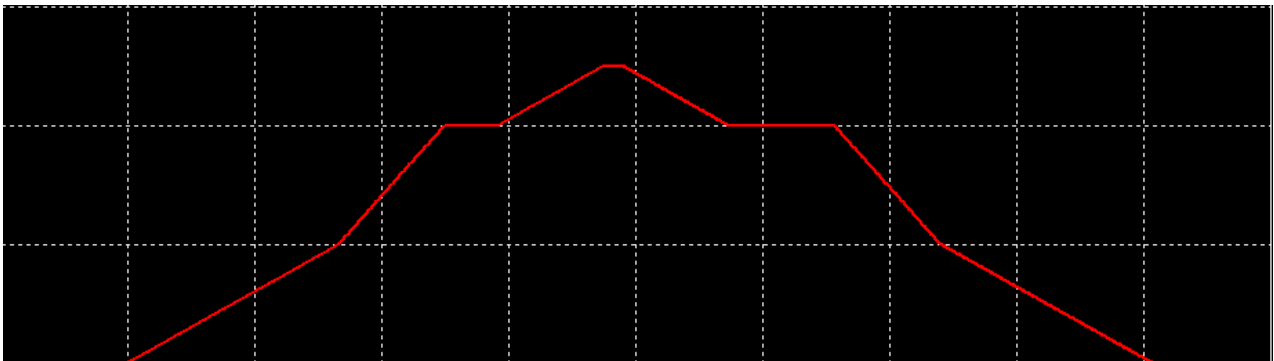
example 4

C# 5

List Motion

× ListMotion

example



C#

```
private void btnTest_Click(object sender, EventArgs e)
{
    //
    // 가 ( 가 ,
    // )
    //
    // , (0~7)
    // 8

    // 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)
    {
        axisMask1 = (uint)(1 << axisID);
        axisMask2 = 0;
    }
    else
    {
        axisMask1 = 0;
        axisMask2 = (uint)(0x01 << (axisID - 32));
    }
}
```



```
    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
    , 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)
        {
            //
            ec.ecmLmSt_GetRunStepInfo(netID, lmMapIndex, ref
runStepCount, ref runStepID, ref runStepState, ref errorCode);
```

```

        // runStepID :          StepID
        // runStepState :      Step      (Ready, Busy,
Paused, Completed)

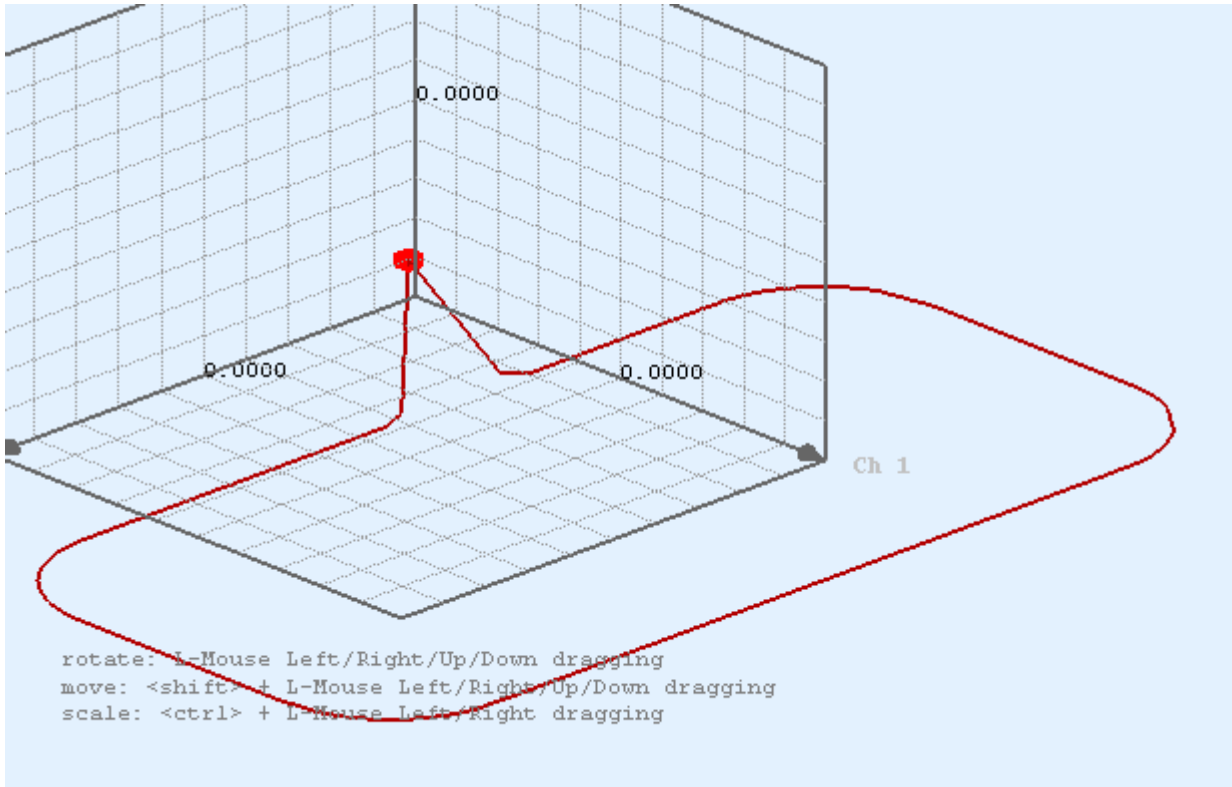
        //          StepID가          StepID      ,      가
Complete
        // StepCount          ecmLmSt_GetRemStepCount
RemStep
        if (runStepID == stepID && runStepState ==
(int)ec.EEcmLmCmdItemSts.ecmLM_CMDITEM_STS_COMPLETED)
        {
            isSuccess = true;
            break;
        }
        //
        // 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)
{
    //
}
}

```

example



C#

```
private void btnTest2_Click(object sender, EventArgs e)
{
    //
    //
    //
    //      8      , (0~7)

    //
    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);

    //
    //      , 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
}, 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)
        {
            //          .
            ec.ecmLmSt_GetRunStepInfo(netID, lmMapIndex, ref
runStepCount, ref runStepID, ref runStepState, ref errorCode);

            // runStepID :          StepID
            // runStepState :          Step          (Ready, Busy,
Paused, Completed)

            //          StepID가          StepID          ,          가
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.co.kr/info/> - -

Permanent link:

http://comizoa.co.kr/info/doku.php?id=platform:ethercat:100_api:20_eg:listmotion

Last update: **2024/07/08 18:23**