

TakTouch 1000 Programming Interface Methods

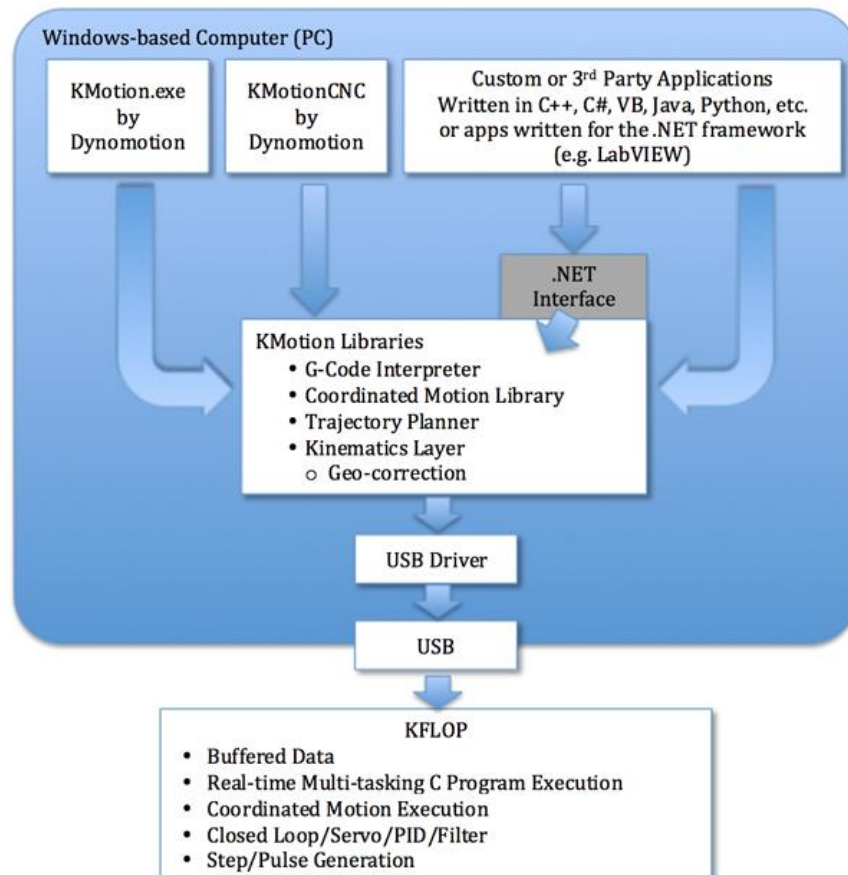
1 Overview

Real time motion control for the TakTouch 1000 is handled with a dedicated DSP motion control board (KFLOP from Dynamotion) which is connected to a host PC via USB (utilizes ASCII [commands](#)).

1.1 The following block diagram shows the software flow from the PC to the KFLOP motion controller which drives the TakTouch. This [link](#) includes more info:

Dynamotion's current software package comprises three major components:

1. **KMotion.exe** (pronounced 'commotion'), which is used to configure settings in KFLOP, initialize axes, and to write C programs to control your system.
2. **KMotionCNC**, which is our CNC application that has a GUI (Graphical User Interface) and is used to control systems based on G-Code. Alternatively, we have a Mach3 plugin so our boards can be used with the Mach3 CNC application.
3. The **KMotion libraries** that form the heart of the other two software components as well as user-created or third-party applications.



2 Available Methods for Programming the TakTouch

2.1 Via .NET

Any software that can access the .NET interface can easily utilize all the libraries and functions for controlling the TakTouch. These include custom or 3rd party applications written in C++, C#, VB, Java, Python etc. and apps written for the .NET framework such as LabVIEW.

2.1.1 The following sample code excerpt shows how to program the TakTouch to perform a simple tap at two XY locations in C#:

```
//Position at X-200mm, Y-200mm
```

```
axisX.MoveTo(-200);
```

```
axisY.MoveTo(-200);
```

```
//Tap once
```

```
axisZ.MoveTo(-25);
```

```
axisZ.MoveTo(0);
```

```
//Move over 100mm in X
```

```
axisX.RelativeMoveTo(100);
```

```
//Tap once
```

```
axisZ.MoveTo(-25);
```

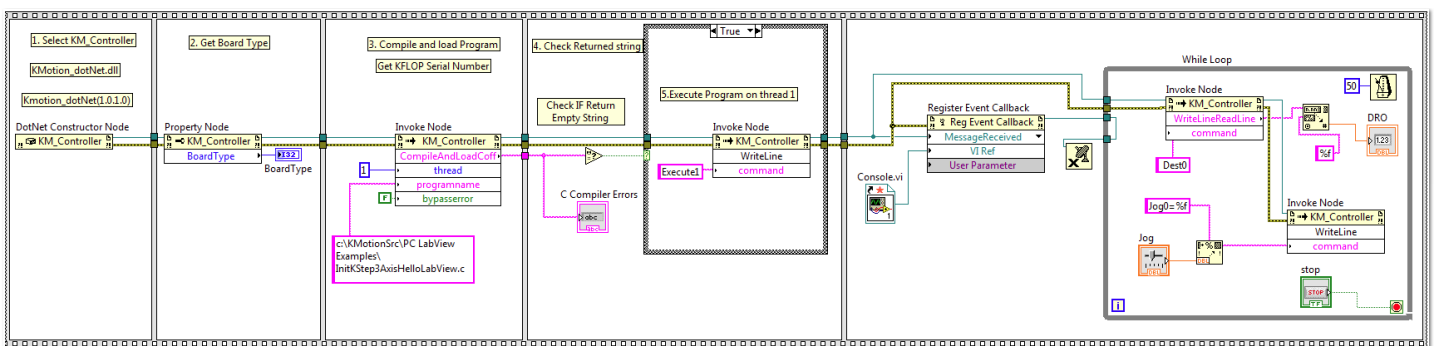
```
axisZ.MoveTo(0);
```

```
//Return to Home
```

```
axisX.MoveTo(0);
```

```
axisY.MoveTo(0);
```

2.1.2 The following LabVIEW diagram demonstrates some code for creating a simple GUI. LabVIEW can access the same .NET libraries and functions as C#:



2.1.3 A Dynamotion/KFLOP + LabVIEW reference video can be viewed [here](#)

2.2 With G-Code (RS-274)

This is the quickest and simplest way to control the TakTouch for most users. A GUI is provided by Dynomotion and execution of multi-axis coordinated moves can be programmed in minutes. The G-code interface also allows buffered (real-time) calls of C-programs and provides methods for passing variables to other applications and generating simple PC commands.

2.2.1 The following sample code excerpt shows how to program the TakTouch to perform a simple tap at two XY locations in G-Code:

(Move to X-200mm Y-200mm and Tap once)

g0 x-200 y-200

g1 z-25 f10000

g0 z0

(Move over 100mm in X and Tap once)

g91 g0 x 100

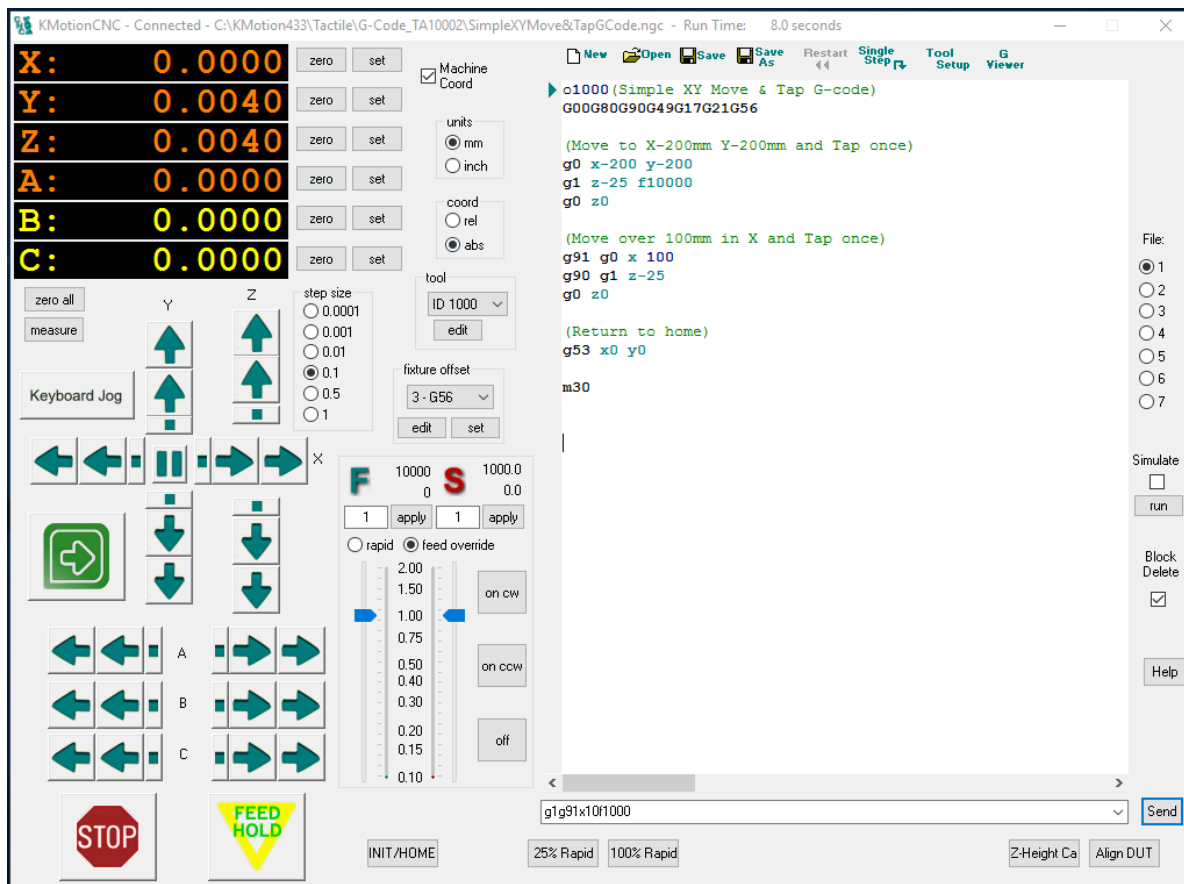
g90 g1 z-25

g0 z0

(Return to home)

g53 x0 y0

2.2.2 GUI for G-Code:



2.3 With C

C programs may be used alone, or in conjunction with G-Code or a .NET application to execute real-time commands and functions directly in the KFLOP motion control board. A GUI provides access to I/O, axis status and setup, an editor/compiler, and button commands.

2.3.1 The following sample code excerpt shows how to program the TakTouch to perform a simple tap at two XY locations using C:

```
main()
{
    //Position at X-200mm, Y-200mm
    Move(X,-200*MM);
    Move(Y,-200*MM);
    //Tap once at 25mm deep
    Tap(-25*MM);

    //Move over 100mm in X
    MoveRel(X,100*MM);
    //Tap once
    Tap(-25*MM);

    //Return to Home
    Move(X,0*MM);
    Move(Y,0*MM);
}

void Tap (double depth)
{
    while(!Check0123Done());
    Move(Z,depth);
    while(!CheckDone(Z));
    Move(Z,0*MM);
    while(!CheckDone(Z));
    return 0;
}
```

2.3.2 GUI for C programs with I/O, axis status, and control:

KMotion - Connected
 File View USB Locations Options Help

Console Axis C Config Bode Analog Step IR Digital G code
 Program & Flash Plot Status Response Filter ID code

Console

```
No tool, ok to proceed
Robot homing calibration process started
Switch #24 checked good
Switch #25 checked good
Switch #34 checked good
Switch #35 checked good
Neg Limit Disabled Axis:4
Robot homing calibration process successful
```

Version: [dropdown] Send
 MoveRel0=100 [dropdown] Send
 SetBit46;ClearBit47 [dropdown] Send
 ReadBit46 [dropdown] Send
 Pos0 [dropdown] Send
 Zero0 [dropdown] Send

Axis

| Dest | Position | Enable | Modes | Done |
|------|----------|--------|--|-------------------------------------|
| #0 | -0.00 | #0 | 0 <input checked="" type="checkbox"/> Encoder CL Step | <input checked="" type="checkbox"/> |
| #1 | -0.00 | #1 | 1 <input checked="" type="checkbox"/> Encoder CL Step | <input checked="" type="checkbox"/> |
| #2 | 0.00 | #2 | 1 <input checked="" type="checkbox"/> Encoder CL Step | <input checked="" type="checkbox"/> |
| #3 | 0.00 | #3 | 0 <input checked="" type="checkbox"/> Encoder CL Step | <input checked="" type="checkbox"/> |
| #4 | 829.27 | #4 | 0 <input type="checkbox"/> No Input Step Dir | <input type="checkbox"/> |
| #5 | 0.00 | #5 | -1 <input type="checkbox"/> Encoder No Output | <input type="checkbox"/> |
| #6 | 0.00 | #6 | 0 <input type="checkbox"/> Encoder No Output | <input type="checkbox"/> |
| #7 | 0.00 | #7 | 1 <input type="checkbox"/> Encoder No Output | <input type="checkbox"/> |

Program - C:\KMotion433\Tactile\C_Prg\SimpleXYMove&Tap.c

```
#define MM 320

void Tap (double depth);

main()
{
  //Position at X=200mm, Y=200mm
  Move (X,-200*MM);
  Move (Y,-200*MM);
  //Tap once at 25mm deep
  Tap (-25*MM);

  //Move over 100mm in X
  MoveRel (X,100*MM);
  //Tap once
  Tap (-25*MM);

  //Return to Home
  Move (X,0*MM);
  Move (Y,0*MM);
}

void Tap (double depth)
{
  while (!Check0123Done());
  Move (Z,depth);
  while (!CheckDone (Z));
  Move (Z,0*MM);
  while (!CheckDone (Z));
  return 0;
}

Thread:
1
2
3
4
5
6
7
```

No Errors, No Warnings, text=9488, bss=0, data=1952, total=11456