## Lektion 6 : Verwendung der Module ti_hub & ti_rover

## Anwendung : Polygone

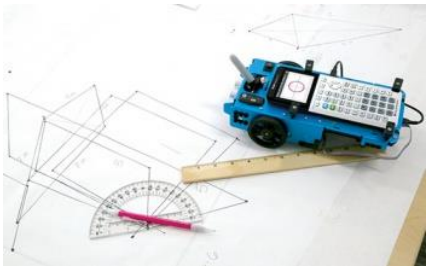
In dieser Anwendung von Lektion 6 soll ein Programm erstellt werden, mit dem der ROVER während einer Fahrt Koordinaten von Punkten aufnimmt, um sie dann grafisch darzustellen. Genutzt werden regelmäßige Vielecke.

**Lernziele :**

- Entwicklung eines Programmes, das die Elemente von **ti_rover** nutzt..
- Verwendung einer For-Schleife.
- Grafische Darstellung von Koordinatenpunkten.

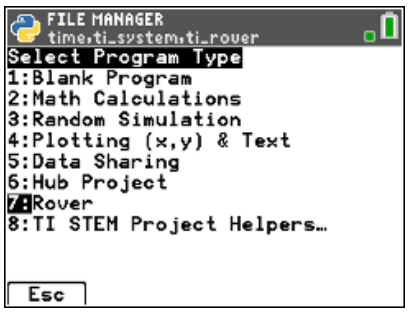
---

Zunächst muss dazu ein Programm geschrieben werden, mit dem der ROVER regelmäßige Vielecke abfährt. Die Koordinaten der Eckpunkte werden in Listen festgehalten, die die Grundlage sind, um eine Zeichnung mittels **ti_plot** zu erstellen. Zugleich werden die Koordinaten in Listen auf dem Taschenrechner gespeichert.

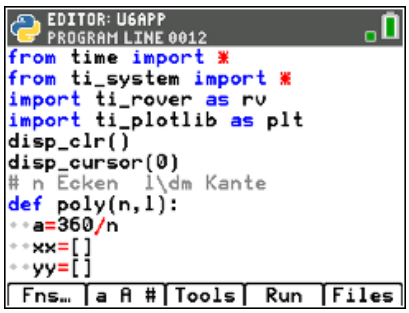


**Das Programm :**

- Anlegen eines neuen Programmes mit dem Namen U6APP.
- Als Typ wird **7 :TI-Rover** verwendet.
- Ferner muss noch das Modul **ti_plotlib** geladen werden.



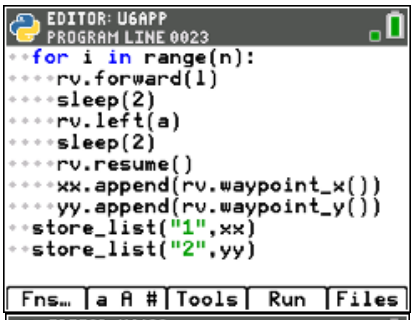

- Obwohl es nicht wirklich notwendig ist, sieht es besser aus, wenn man den Bildschirm löscht und den Cursor ausschaltet. Die Anweisungen **disp_clr()** und **disp_cursor(0)** befinden sich im Modul **ti_system**.
- Anlegen einer Funktion **poly()** mit dem Argument n, der Anzahl der Kanten, sowie l, der Länge einer Kante, in der voreingestellten Einheit von 1dm.
- Der Winkel, um den der ROVER sich an jeder Ecke des Polygons drehen muss, beträgt $a = \frac{360}{n}$ .
- Anlegen der Listen **xx** und **yy** für die Koordinaten der Eckpunkte.

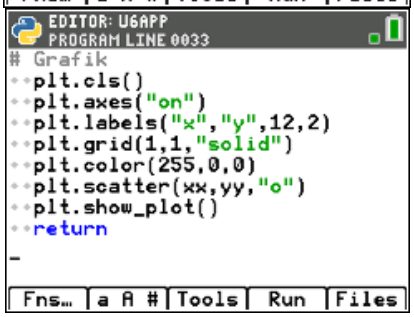


- Anlegen einer **For**-Schleife zur Erstellung des Polygons. Die darin enthaltenen Anweisungen werden n-mal ausgeführt.

Die folgenden Anweisungen befinden sich im Modul **ti_rover.**

- Geradeausfahrt um **l** dm.
- Pause von 2s.
- Linksdrehung um den Winkel **a.**
- Pause von 2s.
- Abspeichern der Koordinaten der Eckpunkte in **xx** und **yy**.
- Verbindung des ROVERs mit dem Taschenrechner aufheben.
- **xx** und **yy** in die Listen $L_1$ und $L_2$ exportieren.



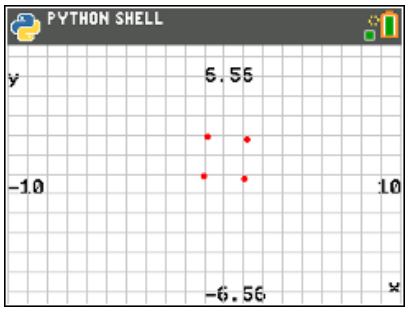




**Lehrertipp** : Die Anweisungen **rv.waypoint_x()** und **rv.waypoint_y()** findet man im Menü **I/O** und dann **3 : Path** im Modul **ti_rover**.

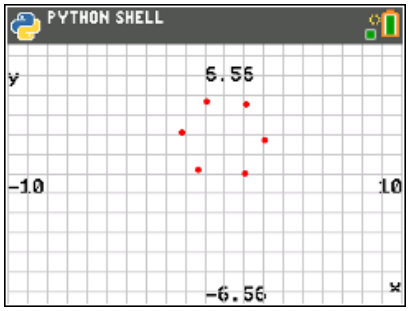
- Erstellen der Grafik wie bisher schon behandelt. Es ist auch eine grafische Darstellung außerhalb Pythons durch die Listen $\mathbf{L_1}$ und $\mathbf{L_2}$ möglich.

**Ergebnis :**

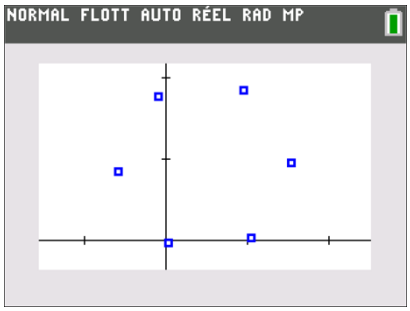
Startet man das Programm und ruft man über [vars] die Funktion **poly( )** mit den Argumenten **poly(4, 2)** auf, so erhält man die Eckpunkte eines Quadrates von 2dm Seitenlänge. Durch Reibung während des Drehens kann die Figur leicht verzerrt aussehen.




So sieht die Grafik bei einem Hexagon aus.



Mit den in den Taschenrechner exportierten Listen lässt sich die nebenstehende Grafik erzeugen, deren Punkte man mit [trace] abtasten und die Koordinaten bestimmen kann.




**Lehrertipp**: Für diese Art von Übung sollte man nicht die Anweisungen rv.pathlist_x () und rv.pathlist_y () verwenden.
Tatsächlich zeichnet der Rechner beim Zeichnen einer Seite des Polygons die Koordinaten der Eckpunkte dieser Strecke auf. Die Koordinaten des letzten Punktes sind also die Koordinaten des ersten Punktes der nächsten Seite. Mit den Anweisungen rv.pathlist_x () und rv.pathlist_y () erhält man die Eckpunktkoordinaten also doppelt.

**Hinweis**: Das Format des Rasters sollte man abhängig von den Polygonen machen, die gezeichnet werden sollen. Hier wurden die Standardeinstellungen verwendet, um die Genauigkeit der ROVER-Spur zu beobachten.

Man sollte auch auf die Art der Oberfläche achten, auf der sich der Roboter bewegt. Diese sollte nicht zu viel Widerstand gegen Bewegung leisten oder im Gegenteil das Rutschen fördern.