**Functions required\_sensor\_assignments.h**

|  |  |
| --- | --- |
| **Syntax** | void assignTurnSensor() |
| **Purpose** | No sensor connected for turning |
| **Entry Conditions** | NONE |
| **Code**  **Example** | //There is no turn sensor to assign  assignTurnSensor(); |

|  |  |
| --- | --- |
| **Syntax** | void assignTurnSensor(int port) |
| **Purpose** | assign a turn sensor. A gyro must be used |
| **Entry Conditions** | * *port*  - port for which the gyro will be connected |
| **Code**  **Example** | //Turn sensor is on analog 1  assignTurnSensor(in1); |

|  |  |
| --- | --- |
| **Syntax** | void assignLiftSensor() |
| **Purpose** | No sensor connected for the lift |
| **Entry Conditions** | NONE |
| **Code**  **Example** | //There is no lift sensor to assign  assignLiftSensor(); |

|  |  |
| --- | --- |
| **Syntax** | void assignLiftSensor(int type, int port) |
| **Purpose** | assign a sensor for the lift |
| **Entry Conditions** | * *type* - the type of sensor that will be used * *port* - the port upon which the sensor is connected |
| **Code**  **Example** | //A potentiometer will be used on analog 2  assignLiftSensor(sensorPotentiometer, in2); |

|  |  |
| --- | --- |
| **Syntax** | void assignLiftSensor(int type, int portMain, int portSub1) |
| **Purpose** | assign a sensor for the lift |
| **Entry Conditions** | * *type* - the type of sensor that will be used * *portMain* - the sensor’s main sensor port * *portSub1* - the sensor’s supporting sensor port |
| **Code**  **Example** | //A quad encoder will be used on digital 1 and digital 3  assignLiftSensor(sensorQuadEncoder, dgtl1, dgtl3); |

|  |  |
| --- | --- |
| **Syntax** | void assignRightDriveSensor() |
| **Purpose** | no sensor connected for right drive |
| **Entry Conditions** | NONE |
| **Code**  **Example** | //There is no sensor connected for the right drive  assignRightDriveSensor(); |

|  |  |
| --- | --- |
| **Syntax** | void assignRightDriveSensor(int type, int port) |
| **Purpose** | assign a sensor for the right drive. Encoders are recommended. |
| **Entry Conditions** | * *type* - the type of sensor that will be used * *port* - the port upon which the sensor is connected |
| **Code**  **Example** | //connect an I2C integrated encoder for the right drive  assignRightDriveSensor(sensorQuadEncoderOnI2CPort, I2C\_1); |

|  |  |
| --- | --- |
| **Syntax** | void assignRightDriveSensor(int type, int portMain,  int portSub1) |
| **Purpose** | assign a sensor for the right drive. Encoders are recommended. |
| **Entry Conditions** | * *type* - the type of sensor that will be used * *portMain* - the main port upon which the sensor is connected * *portSub1* - the sensor’s supporting port |
| **Code**  **Example** | //connect a quad encoder for the right drive  assignRightDriveSensor(sensorQuadEncoder, dgtl1, dgtl2); |

|  |  |
| --- | --- |
| **Syntax** | void assignLeftDriveSensor() |
| **Purpose** | no sensor connected for left drive |
| **Entry Conditions** | NONE |
| **Code**  **Example** | //There is no sensor connected for the left drive  assignRightDriveSensor(); |

|  |  |
| --- | --- |
| **Syntax** | void assignLeftDriveSensor(int type, int port) |
| **Purpose** | assign a sensor for the left drive. Encoders are recommended. |
| **Entry Conditions** | * *type* - the type of sensor that will be used * *port* - the port upon which the sensor is connected |
| **Code**  **Example** | //connect an I2C integrated encoder for the left drive  assignRightDriveSensor(sensorQuadEncoderOnI2CPort, I2C\_2); |

|  |  |
| --- | --- |
| **Syntax** | void assignLeftDriveSensor(int type, int portMain,  int portSub1) |
| **Purpose** | assign a sensor for the left drive. Encoders are recommended. |
| **Entry Conditions** | * *type* - the type of sensor that will be used * *portMain* - the main port upon which the sensor is connected * *portSub1* - the sensor’s supporting port |
| **Code**  **Example** | //connect a quad encoder for the left drive  assignLeftDriveSensor(sensorQuadEncoder, dgtl1, dgtl2); |

|  |  |
| --- | --- |
| **Syntax** | void assignStraffDriveSensor() |
| **Purpose** | no sensor connected for straffing drive |
| **Entry Conditions** | NONE |
| **Code**  **Example** | //There is no sensor connected for the straffing drive  assignStraffDriveSensor(); |

|  |  |
| --- | --- |
| **Syntax** | void assignStraffDriveSensor(int type, int port) |
| **Purpose** | assign a sensor for the straffing drive. Encoders are recommended. |
| **Entry Conditions** | * *type* - the type of sensor that will be used * *port* - the port upon which the sensor is connected |
| **Code**  **Example** | //connect an I2C integrated encoder for the straffing drive  assignStraffDriveSensor(sensorQuadEncoderOnI2CPort, I2C\_1); |

|  |  |
| --- | --- |
| **Syntax** | void assignStraffDriveSensor(int type, int portMain,  int portSub1) |
| **Purpose** | assign a sensor for the straffing drive. Encoders are recommended. |
| **Entry Conditions** | * *type* - the type of sensor that will be used * *portMain* - the main port upon which the sensor is connected * *portSub1* - the sensor’s supporting port |
| **Code**  **Example** | //connect a quad encoder for the straffing drive  assignStraffDriveSensor(sensorQuadEncoder, dgtl1, dgtl2); |