```
// Include sstream for serial parsing
#include <sstream>

// Prototypes for hidden vex functions to bypass PROS bug
extern "C" int32_t vexGenericSerialReceive( uint32_t index, uint8_t
 *buffer, int32_t length );
extern "C" void vexGenericSerialEnable(  uint32_t index, uint32_t
 nu );
extern "C" void vexGenericSerialBaudrate(  uint32_t index, uint32_t
 rate );

// Port to use for serial data
#define SERIALPORT 15           // Port 15

// Variable to put the gyro value into
double gyroValue = 0;

// Currently reads serial data & parses for gyro value
// Can be expanded to look for lidar distance, etc.
void serialRead(void* params) {

    // Start serial on desired port
    vexGenericSerialEnable( SERIALPORT - 1, 0 );

    // Set BAUD rate
    vexGenericSerialBaudrate( SERIALPORT - 1, 115200 );

    // Let VEX OS configure port
    pros::delay(10);

    // Serial message format:
    // D[LIDAR DIST]I[IR DATA]A[GYRO ANGLE]E
    // Example Message:
    // D50.2I128A12.32E

    while (true) {

        // Buffer to store serial data
        uint8_t buffer[256];
        int len = 256;

        // Get serial data
        int32_t nRead = vexGenericSerialReceive(SERIALPORT - 1,
         buffer, len);

        // Now parse the data
        if (nRead >= 9) {

            // Stream to put the characters in
            std::stringstream myStream("");
            bool recordAngle = false;

            // Go through characters
            for (int i = 0; i < nRead; i++) {
                // Get current char
                char thisDigit = (char)buffer[i];
                // If its special, then don't record the value
                if (thisDigit == 'D' || thisDigit == 'I' || thisDigit
                 == 'A')
                    recordAngle = false;
                // Finished recieving angle, so put into variable
                if (thisDigit == 'E') {
                    recordAngle = false;
                    myStream >> gyroValue;
                }
                // If we want the digits, put them into stream
                if (recordAngle)
                    myStream << (char)buffer[i];
                // If the digit is 'A', then the following data is the
                 angle
                if (thisDigit == 'A')
                    recordAngle = true;
            }
        }

        // Delay to let serial data arrive
        pros::delay(10);
    }
}
```