```c
#include <sys/types.h>
#include <sys/socket.h>
#include <stdio.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <stdlib.h>
#include <pthread.h>
#include <string.h>
#include <signal.h>

int sendState = 0;
int sockfd;
pid_t pid;

void *thread_recieveData(void *arg);
void *thread_sendSignal(void *arg);

int main (int argc, char *argv[])
{
  //Socket setup
  int result, len;
  struct sockaddr_in address;
  sockfd = socket(AF_INET, SOCK_STREAM, 0);
  address.sin_family = AF_INET;
  address.sin_addr.s_addr = inet_addr("192.168.0.142");
  address.sin_port = htons(9734);
  len = sizeof(address);
  result=connect(sockfd, (struct sockaddr *) &address, len);
  if (result == -1) {
    perror("oops:␣client2");
    exit(1);
  }
  //

  //Start threads
  pthread_t recieveThread;
  pthread_t sendThread;
  pthread_create(&recieveThread, NULL, thread_recieveData, NULL);
  pthread_create(&sendThread, NULL, thread_sendSignal, NULL);
  int gnuplotCreated;

  for(;;) {
    //IF ladder for choosing what to do
    char input[10];
    fgets(input, 10, stdin);
    if(strcmp(input, "s\n") == 0) {
      sendState = 1;
    } else if(strcmp(input, "q\n") == 0) {
      kill(pid, SIGINT);
      break;
    } else if(strcmp(input, "p\n") == 0 && !gnuplotCreated) {
      gnuplotCreated = 1;
      pid = fork();
      if(pid == 0) {
        execlp("gnuplot", "gnuplot", "plot.p", NULL);
      }
    }

  }
  printf("Closing␣Socket!\n");
  close(sockfd);
  exit(0);
}

//THREAD// recieving data from pi
void *thread_recieveData(void *arg) {
  int recieve;
  FILE *file = fopen("recieve.d", "w");
  for(;;) {
    read(sockfd, &recieve, sizeof(recieve));
    fprintf(file, "%d\n", recieve);
    fflush(file);
  }
  fclose(file);
}

//THREAD// sending signal to pi
void *thread_sendSignal(void *arg) {
  for(;;) {
    if(sendState) {
      printf("%d\n", sendState);
      sendState = 0;
      write(sockfd, "s", 10);
    }
  }
}
```