

<b>Функция</b>	<b>Описание</b>	<b>Время выполнения</b>
<b>void</b> function_send_byte ( <b>char</b> slv_addr, <b>char</b> address, <b>char</b> byte)	Функция записывает один байт (byte) в выбранный регистр (address) ведомого (slv_addr).	При скорости 100 кбит/с: 0.399 мс.  При скорости 400 кбит/с: 0.112 мс.
<b>char</b> function_get_1( <b>char</b> slv_addr, <b>char</b> address)	Функция возвращает один байт, считанный из выбранного регистра (address) ведомого (slv_addr).	При скорости 100 кбит/с: 0.415 мс.  При скорости 400 кбит/с: 0.116 мс.
<b>void</b> function_get_X_accel ( <b>char*</b> mass)	Функция заполняет массив "mass" выходными данными по оси X, считанными с акселерометра. mass[0] - младший байт данных, mass[1] - старший байт.	При скорости 100 кбит/с: 0.504 мс.  При скорости 400 кбит/с: 0.140 мс.
<b>void</b> function_get_Y_accel ( <b>char*</b> mass)	Функция заполняет массив "mass" выходными данными по оси Y, считанными с акселерометра. mass[0] - младший байт данных, mass[1] - старший байт.	При скорости 100 кбит/с: 0.504 мс.  При скорости 400 кбит/с: 0.140 мс.
<b>void</b> function_get_Z_accel ( <b>char*</b> mass)	Функция заполняет массив "mass" выходными данными по оси Z, считанными с акселерометра. mass[0] - младший байт данных, mass[1] - старший байт.	При скорости 100 кбит/с: 0.504 мс.  При скорости 400 кбит/с: 0.140 мс.
<b>void</b> function_get_X_gyro ( <b>char*</b> mass)	Функция заполняет массив "mass" выходными данными по оси X, считанными с гироскопа. mass[0] - младший байт данных, mass[1] - старший байт.	При скорости 100 кбит/с: 0.504 мс.  При скорости 400 кбит/с: 0.140 мс.
<b>void</b> function_get_Y_gyro ( <b>char*</b> mass)	Функция заполняет массив "mass" выходными данными по оси Y, считанными с гироскопа. mass[0] - младший байт данных, mass[1] - старший байт.	При скорости 100 кбит/с: 0.504 мс.  При скорости 400 кбит/с: 0.140 мс.

<b>void</b> function_get_Z_gyro (char* mass)	Функция заполняет массив "mass" выходными данными по оси Z, считанными с гироскопа. mass[0] - младший байт данных, mass[1] - старший байт.	При скорости 100 кбит/с: 0.504 мс.  При скорости 400 кбит/с: 0.140 мс.
<b>void</b> function_get_X_magnet (char* mass)	Функция заполняет массив "mass" выходными данными по оси X, считанными с магнитометра. mass[0] - младший байт данных, mass[1] - старший байт.	При скорости 100 кбит/с: 0.506 мс.  При скорости 400 кбит/с: 0.143 мс.
<b>void</b> function_get_Y_magnet (char* mass)	Функция заполняет массив "mass" выходными данными по оси Y, считанными с магнитометра. mass[0] - младший байт данных, mass[1] - старший байт.	При скорости 100 кбит/с: 0.506 мс.  При скорости 400 кбит/с: 0.143 мс.
<b>void</b> function_get_Z_magnet (char* mass)	Функция заполняет массив "mass" выходными данными по оси Z, считанными с магнитометра. mass[0] - младший байт данных, mass[1] - старший байт.	При скорости 100 кбит/с: 0.506 мс.  При скорости 400 кбит/с: 0.143 мс.
<b>void</b> function_get_X_Y_Z (char* mass, char slv_addr, char address)	Функция заполняет массив "mass" выходными данными по выбранной оси (в качестве переменной address указывается номер первого регистра, с которого начинается чтение), считанными с выбранного устройства (slv_addr). mass[0] - младший байт данных, mass[1] - старший байт.	<i>Магнитометр:</i> При скорости 100 кбит/с: 0.508 мс.  При скорости 400 кбит/с: 0.143 мс.  <i>Аксел. и гироскоп:</i> При скорости 100 кбит/с: 0.507 мс.  При скорости 400 кбит/с: 0.141 мс.
<b>void</b> function_get_XYZ_6_gyro (char* mass)	Функция заполняет массив "mass" выходными данными со всех трех осей гироскопа. Данные расположены в следующем порядке: X_L X_H Y_L Y_H Z_L Z_H.	При скорости 100 кбит/с: 0.874 мс.  При скорости 400 кбит/с:

		0.246 мс.
<b>void</b> function_get_XYZ_6_accel ( <b>char*</b> mass)	Функция заполняет массив "mass" выходными данными со всех трех осей акселерометра. Данные расположены в следующем порядке: X_L X_H Y_L Y_H Z_L Z_H.	При скорости 100 кбит/с: 0.874 мс.  При скорости 400 кбит/с: 0.246 мс.
<b>void</b> function_get_XYZ_6_magnet ( <b>char*</b> mass)	Функция заполняет массив "mass" выходными данными со всех трех осей магнитометра. Данные расположены в следующем порядке: X_L X_H Y_L Y_H Z_L Z_H.	При скорости 100 кбит/с: 0.881 мс.  При скорости 400 кбит/с: 0.253 мс.
<b>void</b> function_get_XYZ_6( <b>char*</b> mass, <b>char</b> slv_addr)	Функция заполняет массив "mass" выходными данными со всех трех осей устройства, адрес которого указывается в качестве переменной slv_addr. Данные расположены в следующем порядке: X_L X_H Y_L Y_H Z_L Z_H.	<i>Магнитометр:</i> При скорости 100 кбит/с: 0.887 мс.  При скорости 400 кбит/с: 0.254 мс.  <i>Аксел. и гироскоп:</i> При скорости 100 кбит/с: 0.878 мс.  При скорости 400 кбит/с: 0.246 мс.
<b>int</b> Filter ( <b>signed int</b> *now, <b>signed int</b> *last, <b>int</b> num_FRAC)	На вход функции подается шесть байт: 2 байта - массив "now" (полученные данные: now[0] - младший байт данных, now[1] - старший байт), 4 байта - массив "last" (данные, полученные на предыдущих итерациях). Функция возвращает отфильтрованное значение (2 байта), а также записывает в массив "last" 4 байта, которые могут быть использованы при дальнейшей фильтрации в качестве данных, полученных на предыдущих итерациях (старший байт - растянутый бит знака из 7 бита третьего байта, третий и второй байты - отфильтрованные выходные данные, первый байт - байт FRAC). Переменная "num_FRAC" определяет глубину фильтрации. Данные в массиве "now" расположены в следующем порядке: now[0]=(растянутый_байт_знака;X_H;X_L;00), now[1]=(растянутый_байт_знака;Y_H;Y_L;00), now[2]=(растянутый_байт_знака;Z_H;Z_L;00). Данные в массиве "last" расположены	4.35 мкс.

	аналогичным образом за исключением младшего байта, который заполнен не нулями, а представляет собой байт FRAC (используется при дальнейшей фильтрации).	
<b>void</b> function_get_XYZ_6_gyro_filter( <b>int</b> *mass, <b>signed int</b> *now, <b>signed int</b> *last, <b>char</b> choice_cycle, <b>int</b> num_FRAC)	Функция принимает шесть байт данных от гироскопа, записывает их в массив "now", фильтрует и записывает в массив "last". Если переменная choice_cycle = 0, то функция самостоятельно заполнит массив "last", приняв данные и, после паузы, заполнит массив "now", приняв данные повторно, иначе (choice_cycle = 1) для фильтрации используются данные, записанные пользователем в массив "last", а массив "now" заполняется текущими данными. Переменная "num_FRAC" определяет глубину фильтрации. Данные в массиве "now" расположены в следующем порядке: now[0]=(растянутый_байт_знака;X_H;X_L;00), now[1]=(растянутый_байт_знака;Y_H;Y_L;00), now[2]=(растянутый_байт_знака;Z_H;Z_L;00). Данные в массиве "last" расположены аналогичным образом за исключением младшего байта, который не заполнен нулями, а представляет собой байт FRAC (используется при дальнейшей фильтрации). Период обновления выходных данных = 5 мс (200 Hz). Массив "mass" должен содержать шесть элементов.	<i>При скорости 100 кбит/с:</i> При choice_cycle = 0: 6.768 мс. При choice_cycle = 1: 0.895 мс.  <i>При скорости 400 кбит/с:</i> При choice_cycle = 0: 5.509 мс. При choice_cycle = 1: 0.266 мс.
<b>void</b> function_get_XYZ_6_accel_filter( <b>int</b> *mass, <b>signed int</b> *now, <b>signed int</b> *last, <b>char</b> choice_cycle, <b>int</b> num_FRAC)	Функция принимает шесть байт данных от акселерометра, записывает их в массив "now", фильтрует и записывает в массив "last". Если переменная choice_cycle = 0, то функция самостоятельно заполнит массив "last", приняв данные и, после паузы, заполнит массив "now", приняв данные повторно, иначе (choice_cycle = 1) для фильтрации используются данные, записанные пользователем в массив "last", а массив "now" заполняется текущими данными. Переменная "num_FRAC" определяет глубину фильтрации. Данные в массиве "now" расположены в следующем порядке: now[0]=(растянутый_байт_знака;X_H;X_L;00), now[1]=(растянутый_байт_знака;Y_H;Y_L;00), now[2]=(растянутый_байт_знака;Z_H;Z_L;00). Данные в массиве "last" расположены аналогичным образом за исключением младшего байта, который не заполнен нулями, а представляет собой байт FRAC (используется при дальнейшей фильтрации). Период обновления выходных данных = 5 мс (200 Hz). Массив "mass" должен содержать шесть элементов.	<i>При скорости 100 кбит/с:</i> При choice_cycle = 0: 6.768 мс. При choice_cycle = 1: 0.895 мс.  <i>При скорости 400 кбит/с:</i> При choice_cycle = 0: 5.509 мс. При choice_cycle = 1: 0.266 мс.

<p><b>void</b> function_get_XYZ_6_magnet_filter(<b>int</b> *mass, <b>signed int</b> *now, <b>signed int</b> *last, <b>char</b> choice_cycle, <b>int</b> num_FRAC)</p>	<p>Функция принимает шесть байт данных от магнитометра, записывает их в массив "now", фильтрует и записывает в массив "last". Если переменная choice_cycle = 0, то функция самостоятельно заполнит массив "last", приняв данные и, после паузы, заполнит массив "now", приняв данные повторно, иначе (choice_cycle = 1) для фильтрации используются данные, записанные пользователем в массив "last", а массив "now" заполняется текущими данными. Переменная "num_FRAC" определяет глубину фильтрации. Данные в массиве "now" расположены в следующем порядке: now[0]=(растянутый_байт_знака;X_H;X_L;00), now[1]=(растянутый_байт_знака;Y_H;Y_L;00), now[2]=(растянутый_байт_знака;Z_H;Z_L;00). Данные в массиве "last" расположены аналогичным образом за исключением младшего байта, который не заполнен нулями, а представляет собой байт FRAC (используется при дальнейшей фильтрации). Период обновления выходных данных = 5 мс (200 Hz). Массив "mass" должен содержать шесть элементов.</p>	<p><i>При скорости 100 кбит/с:</i> При choice_cycle = 0: 6.778 мс. При choice_cycle = 1: 0.906 мс.</p> <p><i>При скорости 400 кбит/с:</i> При choice_cycle = 0: 5.518 мс. При choice_cycle = 1: 0.274 мс.</p>
<p><b>void</b> function_get_XYZ_6_filter(<b>char</b> slv_addr, <b>int</b> *mass, <b>signed int</b> *now, <b>signed int</b> *last, <b>char</b> choice_cycle, <b>int</b> num_FRAC)</p>	<p>Функция принимает шесть байт данных от устройства по указанному адресу (slv_addr), записывает их в массив "now", фильтрует и записывает в массив "last". Если переменная choice_cycle = 0, то функция самостоятельно заполнит массив "last", приняв данные и, после паузы, заполнит массив "now", приняв данные повторно, иначе (choice_cycle = 1) для фильтрации используются данные, записанные пользователем в массив "last", а массив "now" заполняется текущими данными. Переменная "num_FRAC" определяет глубину фильтрации. Данные в массиве "now" расположены в следующем порядке: now[0]=(растянутый_байт_знака;X_H;X_L;00), now[1]=(растянутый_байт_знака;Y_H;Y_L;00), now[2]=(растянутый_байт_знака;Z_H;Z_L;00). Данные в массиве "last" расположены аналогичным образом за исключением младшего байта, который не заполнен нулями, а представляет собой байт FRAC (используется при дальнейшей фильтрации). Период обновления выходных данных = 5 мс (200 Hz). Массив "mass" должен содержать шесть элементов.</p>	<p><i>Магнитометр:</i> <i>При скорости 100 кбит/с:</i> При choice_cycle = 0: 6.782 мс. При choice_cycle = 1: 0.908 мс.</p> <p><i>При скорости 400 кбит/с:</i> При choice_cycle = 0: 5.522 мс. При choice_cycle = 1: 0.277 мс.</p> <p><i>Аксел. и гироскоп:</i> <i>При скорости 100 кбит/с:</i> При choice_cycle = 0: 6.769 мс. При choice_cycle = 1: 0.891 мс.</p> <p><i>При скорости 400 кбит/с:</i> При choice_cycle = 0: 5.513 мс.</p>

		При choice_cycle = 1: 0.267 мс .
<b>void</b> function_get_XYZ_18 ( <b>char</b> *mass)	Функция заполняет массив "mass" выходными данными со всех устройств (18 байт): (0-5) - выходные данные с гироскопа, (6-11) - выходные данные с акселерометра, (12-17) - выходные данные с магнитометра. Данные расположены в следующем порядке: X_L X_H Y_L Y_H Z_L Z_H.	При скорости 100 кбит/с: 2.635 мс.  При скорости 400 кбит/с: 0.750 мс.
<b>void</b> function_get_XYZ_18_filter( <b>int</b> *mass, <b>int</b> *last, <b>int</b> *now, <b>char</b> choice_cycle, <b>int</b> num_FRAC)	Функция принимает 18 байт данных от трех устройств, записывает их в массив "now" (now[0-2] - выходные данные с гироскопа, now[3-5] - выходные данные с акселерометра, now[6-8] - выходные данные с магнитометра), фильтрует и записывает в массив "last" (в аналогичном порядке). Если переменная choice_cycle = 0, то функция самостоятельно заполнит массив "last", приняв данные и, после паузы, заполнит массив "now", приняв данные повторно, иначе (choice_cycle = 1) для фильтрации используются данные, записанные пользователем в массив "last", а массив "now" заполняется текущими данными. Переменная "num_FRAC" определяет глубину фильтрации. Данные в массиве "now" расположены в следующем порядке: now[0]=(растянутый_байт_знака;X_H;X_L;00), now[1]=(растянутый_байт_знака;Y_H;Y_L;00) now[2]=(растянутый_байт_знака;Z_H;Z_L;00). Данные в массиве "last" расположены аналогичным образом, только младший байт не заполнен нулями, а представляет собой байт FRAC (используется при дальнейшей фильтрации). Массив "mass" должен содержать 18 элементов.	<i>При скорости 100 кбит/с:</i> При choice_cycle = 0: 10.342 мс. При choice_cycle = 1: 2.690 мс.  <i>При скорости 400 кбит/с:</i> При choice_cycle = 0: 6.554 мс. При choice_cycle = 1: 0.806 мс.
<b>void</b> function_get_XYZ_6_gyro_f ilter_buf( <b>int</b> *mass, <b>signed</b> <b>int</b> *now, <b>signed int</b> *last, <b>char</b> choice_cycle, <b>int</b> num_FRAC, <b>int</b> itter)	Функция принимает шесть байт данных от гироскопа, записывает их в массив "now", фильтрует и записывает в массив "last". Если переменная choice_cycle = 0, то функция самостоятельно заполнит массив "last", приняв данные и, после паузы, заполнит массив "now", приняв данные повторно, иначе (choice_cycle = 1) для фильтрации используются данные, записанные пользователем в массив "last", а массив "now" заполняется текущими данными. Переменная "num_FRAC" определяет глубину фильтрации. С помощью переменной itter задается количество раз, которое необходимо выполнить цикл (получить новые данные и отфильтровать). После выполнения каждого цикла, принятые данные записываются в массив "now", а отфильтрованные в массив "last" в	<b>Гироскоп (скорость обновления выходных данных – 800 Гц): При itter = 2:</b>  <i>При скорости 100 кбит/с:</i> При choice_cycle = 0: 6.806 мс. При choice_cycle = 1: 4.685 мс.  <i>При скорости 400 кбит/с:</i> При choice_cycle = 0: 4.623 мс.

	<p>следующем формате: имя_массива[0] = (00;00;X_H;X_L), имя_массива[1] = (00;00;Y_H;Y_L), имя_массива[2] = (00;00;Z_H;Z_L) и т.д. Очевидно, что размеры массивов "last" и "now" выбираются исходя из значения переменной itter (размер массива = 3*itter). Массив "mass" должен содержать 6 элементов.</p>	<p>При chose_cycle = 1: 3.127 мс.</p> <p><b>При itter = 1:</b> <i>При скорости 100 кбит/с:</i> При chose_cycle = 0: 4.675 мс. При chose_cycle = 1: 2.555 мс.</p> <p><i>При скорости 400 кбит/с:</i> При chose_cycle = 0: 3.112 мс. При chose_cycle = 1: 1.623 мс.</p>
<p><b>void</b> function_get_XYZ_6_accel_filter_buf(<b>int</b> *mass, <b>signed int</b> *now, <b>signed int</b> *last, <b>char</b> choice_cycle, <b>int</b> num_FRAC, <b>int</b> itter)</p>	<p>Функция принимает шесть байт данных от акселерометра, записывает их в массив "now", фильтрует и записывает в массив "last". Если переменная chose_cycle = 0, то функция самостоятельно заполнит массив "last", приняв данные и, после паузы, заполнит массив "now", приняв данные повторно, иначе (chose_cycle = 1) для фильтрации используются данные, записанные пользователем в массив "last", а массив "now" заполняется текущими данными. Переменная "num_FRAC" определяет глубину фильтрации. С помощью переменной itter задается количество раз, которое необходимо выполнить цикл (получить новые данные и отфильтровать). После выполнения каждого цикла, принятые данные записываются в массив "now", а отфильтрованные в массив "last" в следующем формате: имя_массива[0] = (00;00;X_H;X_L), имя_массива[1] = (00;00;Y_H;Y_L), имя_массива[2] = (00;00;Z_H;Z_L) и т.д. Очевидно, что размеры массивов "last" и "now" выбираются исходя из значения переменной itter (размер массива = 3*itter). Массив "mass" должен содержать 6 элементов.</p>	<p><b>Акселерометр (скорость обновления выходных данных – 1000 Гц):</b> <b>При itter = 2:</b></p> <p><i>При скорости 100 кбит/с:</i> При chose_cycle = 0: 6.060 мс. При chose_cycle = 1: 4.188 мс.</p> <p><i>При скорости 400 кбит/с:</i> При chose_cycle = 0: 3.874 мс. При chose_cycle = 1: 2.628 мс.</p> <p><b>При itter = 1:</b> <i>При скорости 100 кбит/с:</i> При chose_cycle = 0: 4.174 мс. При chose_cycle = 1: 2.305 мс.</p> <p><i>При скорости 400 кбит/с:</i> При chose_cycle = 0: 2.570 мс.</p>

<p><b>void</b> function_get_XYZ_6_magnet_filter_buf(<b>int</b> *mass, <b>signed int</b> *now, <b>signed int</b> *last, <b>char</b> choice_cycle, <b>int</b> num_FRAC, <b>int</b> itter)</p>	<p>Функция принимает шесть байт данных от магнитометра, записывает их в массив "now", фильтрует и записывает в массив "last". Если переменная choice_cycle = 0, то функция самостоятельно заполнит массив "last", приняв данные и, после паузы, заполнит массив "now", приняв данные повторно, иначе (choice_cycle = 1) для фильтрации используются данные, записанные пользователем в массив "last", а массив "now" заполняется текущими данными. Переменная "num_FRAC" определяет глубину фильтрации. С помощью переменной itter задается количество раз, которое необходимо выполнить цикл (получить новые данные и отфильтровать). После выполнения каждого цикла, принятые данные записываются в массив "now", а отфильтрованные в массив "last" в следующем формате: имя_массива[0] = (00;00;X_H;X_L), имя_массива[1] = (00;00;Y_H;Y_L), имя_массива[2] = (00;00;Z_H;Z_L) и т.д. Очевидно, что размеры массивов "last" и "now" выбираются исходя из значения переменной itter (размер массива = 3*itter). Массив "mass" должен содержать 6 элементов.</p>	<p>При choice_cycle = 1: 1.374 мс.</p> <p><b>Магнитометр (скорость обновления выходных данных – 220 Гц):</b> <b>При itter = 2:</b></p> <p>При скорости 100 кбит/с: При choice_cycle = 0: 18.060 мс. При choice_cycle = 1: 12.184 мс.</p> <p>При скорости 400 кбит/с: При choice_cycle = 0: 15.874 мс. При choice_cycle = 1: 10.629 мс.</p> <p><b>При itter = 1:</b> При скорости 100 кбит/с: При choice_cycle = 0: 10.884 мс. При choice_cycle = 1: 6.306 мс.</p> <p>При скорости 400 кбит/с: При choice_cycle = 0: 10.258 мс. При choice_cycle = 1: 5.374 мс.</p>
<p><b>void</b> function_get_XYZ_6_filter_buf(<b>int</b> *mass, <b>signed int</b> *now, <b>signed int</b> *last, <b>char</b> slv_addr, <b>char</b> choice_cycle, <b>int</b> num_FRAC, <b>int</b> itter)</p>	<p>Функция принимает шесть байт данных от выбранного устройства (slv_addr), записывает их в массив "now", фильтрует и записывает в массив "last". Если переменная choice_cycle = 0, то функция самостоятельно заполнит массив "last", приняв данные и, после паузы, заполнит массив "now", приняв данные повторно, иначе (choice_cycle = 1) для фильтрации используются данные, записанные пользователем в массив "last", а массив "now" заполняется текущими данными. Переменная "num_FRAC" определяет глубину фильтрации. С помощью переменной itter задается количество раз, которое необходимо выполнить цикл (получить новые данные и отфильтровать). После выполнения</p>	<p><b>Гироскоп (скорость обновления выходных данных – 800 Гц):</b> <b>При itter = 2:</b></p> <p>При скорости 100 кбит/с: При choice_cycle = 0: 6.806 мс. При choice_cycle = 1: 4.685 мс.</p> <p>При скорости 400 кбит/с:</p>

	<p>каждого цикла, принятые данные записываются в массив "now", а отфильтрованные в массив "last" в следующем формате: имя_массива[0] = (00;00;X_H;X_L), имя_массива[1] = (00;00;Y_H;Y_L), имя_массива[2] = (00;00;Z_H;Z_L) и т.д. Очевидно, что размеры массивов "last" и "now" выбираются исходя из значения переменной itter (размер массива = 3*itter).</p>	<p>При chose_cycle = 0: 4.623 мс. При chose_cycle = 1: 3.127 мс.</p> <p><b>При itter = 1:</b> <i>При скорости 100 кбит/с:</i> При chose_cycle = 0: 4.675 мс. При chose_cycle = 1: 2.555 мс.</p> <p><i>При скорости 400 кбит/с:</i> При chose_cycle = 0: 3.112 мс. При chose_cycle = 1: 1.623 мс</p> <p><b>Акселерометр (скорость обновления выходных данных – 1000 Гц):</b> <b>При itter = 2:</b></p> <p><i>При скорости 100 кбит/с:</i> При chose_cycle = 0: 6.060 мс. При chose_cycle = 1: 4.188 мс.</p> <p><i>При скорости 400 кбит/с:</i> При chose_cycle = 0: 3.874 мс. При chose_cycle = 1: 2.628 мс.</p> <p><b>При itter = 1:</b> <i>При скорости 100 кбит/с:</i> При chose_cycle = 0: 4.174 мс. При chose_cycle = 1: 2.305 мс.</p> <p><i>При скорости 400 кбит/с:</i> При chose_cycle = 0:</p>
--	--	--

2.570 мс.  
При chose\_cycle = 1:  
1.374 мс.

**Магнитометр  
(скорость  
обновления  
выходных данных –  
220 Гц):  
При itter = 2:**

*При скорости 100  
кбит/с:*  
При chose\_cycle = 0:  
18.060 мс.  
При chose\_cycle = 1:  
12.184 мс.

*При скорости 400  
кбит/с:*  
При chose\_cycle = 0:  
15.874 мс.  
При chose\_cycle = 1:  
10.629 мс.

**При itter = 1:**  
*При скорости 100  
кбит/с:*  
При chose\_cycle = 0:  
10.884 мс.  
При chose\_cycle = 1:  
6.306 мс.

*При скорости 400  
кбит/с:*  
При chose\_cycle = 0:  
10.258 мс.  
При chose\_cycle = 1:  
5.374 мс.