Wait, microcontroller?

Arduino?!

```c
 */
static void __do_notify(struct mqueue_inode_info *info)
{
        /* notification
         * invoked when there is registered process and there isn't process
         * waiting synchronously for message AND state of queue changed from
         * empty to not empty. Here we are sure that no one is waiting
         * synchronously. */
        if (info->notify_owner &&
            info->attr.mq_curmsgs == 1) {
                struct siginfo sig_i;
                switch (info->notify.sigev_notify) {
                case SIGEV_NONE:
                        break;
                case SIGEV_SIGNAL:
                        /* sends signal */

                        sig_i.si_signo = info->notify.sigev_signo;
                        sig_i.si_errno = 0;
                        sig_i.si_code = SI_MESGQ;
                        sig_i.si_value = info->notify.sigev_value;
                        /* map current pid/uid into info->owner's namespaces */
                        rcu_read_lock();
                        sig_i.si_pid = task_tgid_nr_ns(current,
                                                ns_of_pid(info->notify_owner));
                        sig_i.si_uid = from_kuid_munged(info->notify_user_ns, current_uid());
                        rcu_read_unlock();
```