

Proposta de Desenvolvimento de um Sistema de Aquecimento para Beteiras

Rafael Feijó Leonardo
Graduação em Engenharia Eletrônica
Universidade de Brasília
Gama, DF, Brasil
goldcard99@hotmail.com

Resumo—Este documento apresenta a proposta de desenvolvimento de um sistema de controle de temperatura *PID* para aquários de pequeno porte, utilizando uma *FPGA* para a correção do erro da temperatura *setpoint*.

I. JUSTIFICATIVA

Com a evolução do aquarismo, eletrônicos diversos foram desenvolvidos buscando reproduzir o *habitat* natural dos peixes.

Dentre estes dispositivos, o aquecedor se tornou um dos mais populares. Isso devido ao fato de que a temperatura corporal do peixe é determinada pela água ao seu redor e, esta por sua vez, determina o metabolismo do animal. Ou seja, para baixas temperaturas o metabolismo desacelera, reduzindo o apetite, que por sua vez baixa a imunidade do peixe, dentre outros problemas característicos. [1]

Entretanto, dentre os diversos modelos de aquecedores disponíveis no mercado, para aquários de até 25 Litros é difícil achar modelos de controle automático. Ou seja, a maioria dos disponíveis devem ser utilizados com um termômetro, em que o usuário deixa aquecer até a temperatura desejada e desliga o aparelho.

Nesse contexto, essa proposta visa desenvolver um sistema automático de controle, utilizando a correção por *PID* [2], para manutenção da temperatura da água de pequenos aquários, utilizando uma placa *FPGA Basys 3*, da *Xilinx* além de dispositivos sensor e atuador.

II. OBJETIVOS

Como objetivo, tem-se o desenvolvimento de um sistema de aquecimento, para aclimação de aquários de até 25 Litros, utilizando um algoritmo *PID*, em tempo real, embarcado em uma *FPGA*.

III. REQUISITOS FUNCIONAIS

- 1) Configuração da Temperatura *Setpoint*
O sistema deverá permitir a configuração da temperatura *setpoint* através de uma palavra de 2 bits, totalizando 4 possíveis temperaturas (24, 25, 26 e 27 °C).
- 2) Indicador *On/Off*
O sistema deverá possuir um *LED* indicativo do estado do sistema: ligado ou desligado.

- 3) Indicador *IDLE/WORKING*
O sistema deverá possuir um *LED* indicativo do estado de controle: *setpoint* atingido (*IDLE*) ou aquecendo (*WORKING*).
- 4) Cálculo *PID*
O sistema deverá ser capaz de realizar o cálculo do erro proporcional, integrativo e derivativo da temperatura recebida do microcontrolador.
- 5) Configuração do Ganho *Kp*, *Ki* e *Kd*
O sistema deverá permitir a configuração dos ganhos das parcelas proporcional, integrativa e derivativa do algoritmo.
- 6) Visualização da Temperatura Real
O sistema deverá apresentar a temperatura relativa do aquário, em tempo real, nos *displays 7 segmentos* da placa *FPGA*.
- 7) *Low Power Mode*
O sistema deverá possuir um botão que, se pressionado, desativará o algoritmo *PID* e colocará o microcontrolador em modo de baixo consumo.

IV. REQUISITOS NÃO FUNCIONAIS

- 1) Sensor de Temperatura
Sensor *DS18B20*, com interface *OneWire*, para aferição da temperatura relativa da água do aquário.
- 2) Microcontrolador
Microcontrolador *NodeMCU*, como ponte entre o sensor de temperatura e a placa *FPGA*, através de comunicação serial.
- 3) *FPGA*
Placa *Basys 3* responsável pelo cálculo do algoritmo *PID* e controle do *driver* do aquecedor.
- 4) Aquecedor
Pastilha *Peltier* como componente aquecedor.

V. DESCRIÇÃO DO MODELO

A modelagem do sistema em *hardware* é descrita pelas Figuras 1 e 2.

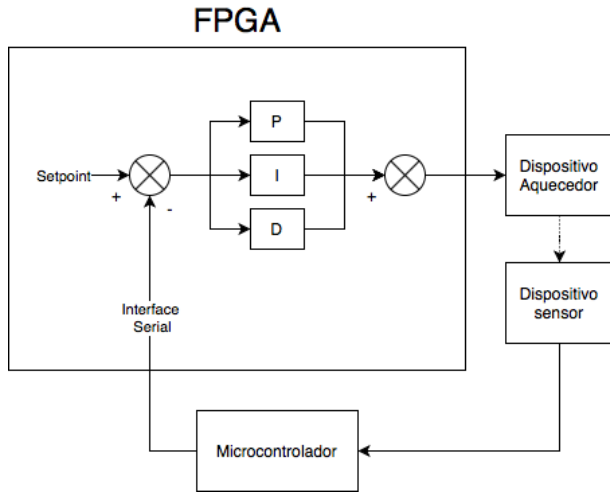


Figura 1. Diagrama modular do sistema de controle proposto. [2]

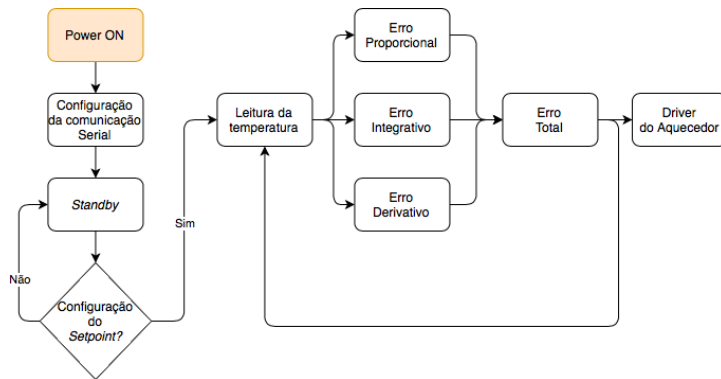


Figura 2. Fluxograma geral do sistema de controle proposto. [3]

VI. METODOLOGIA

Como metodologia, o projeto foi dividido em 6 etapas, descritas abaixo.

- 1) Projeto de *Hardware* para conexão do microcontrolador, sensor, aquecedor e *FPGA*.
- 2) Aquisição de dados de temperatura e interface serial (*FPGA* - Microcontrolador).
- 3) Algoritmo PID: Bloco Proporcional. [4]
- 4) Algoritmo PID: Bloco Integrador. [4]
- 5) Algoritmo PID: Bloco derivativo. [4]
- 6) Integração entre os blocos e *driver* do aquecedor.

Em cada etapa, serão realizadas simulações comportamentais que comprovem o funcionamento do bloco para posterior implementação em *hardware*.

VII. CRONOGRAMA

Tabela I
CRONOGRAMA PROPOSTO PARA O PROJETO.

| Semana | Data | PC | Meta |
|--------|---------------|----|------|
| 0 | 12/10 à 16/10 | X | 1 |
| 1 | 19/10 à 23/10 | | 2 |
| 2 | 26/10 à 30/10 | | 3 |
| 3 | 02/11 à 06/11 | | 3 |
| 4 | 09/11 à 13/11 | X | 4 |
| 5 | 16/11 à 20/11 | | 4 |
| 6 | 23/11 à 27/11 | | 5 |
| 7 | 30/11 à 04/12 | | 5 |
| 8 | 07/12 à 11/12 | | 6 |
| 9 | 14/12 à 18/12 | X | - |

REFERÊNCIAS

- [1] “A temperatura da água é importante para o crescimento dos peixes?” Disponível em <https://www.cpt.com.br/cursos-criacaodepeixes/artigos/a-temperatura-da-agua-e-importante-para-o-crescimento-dos-peixes> (acesso em: 06/10/2020).
- [2] C. M. e. J. C. Juan Ordóñez, “Control system in open-source fpga for a self-balancing robot,” Disponível em shorturl.at/cGP79 (acesso em: 10/10/2020).
- [3] Wikipédia, “Controlador proporcional integral derivativo,” Disponível em https://pt.wikipedia.org/wiki/Controlador_proporcional_integral_derivativo (acesso em: 09/10/2020).
- [4] D. Chakraborty, “Pid controller using fpga,” Disponível em <https://github.com/deepc94/pid-fpga-vhdl> (acesso em: 08/10/2020).