# Technická dokumentace

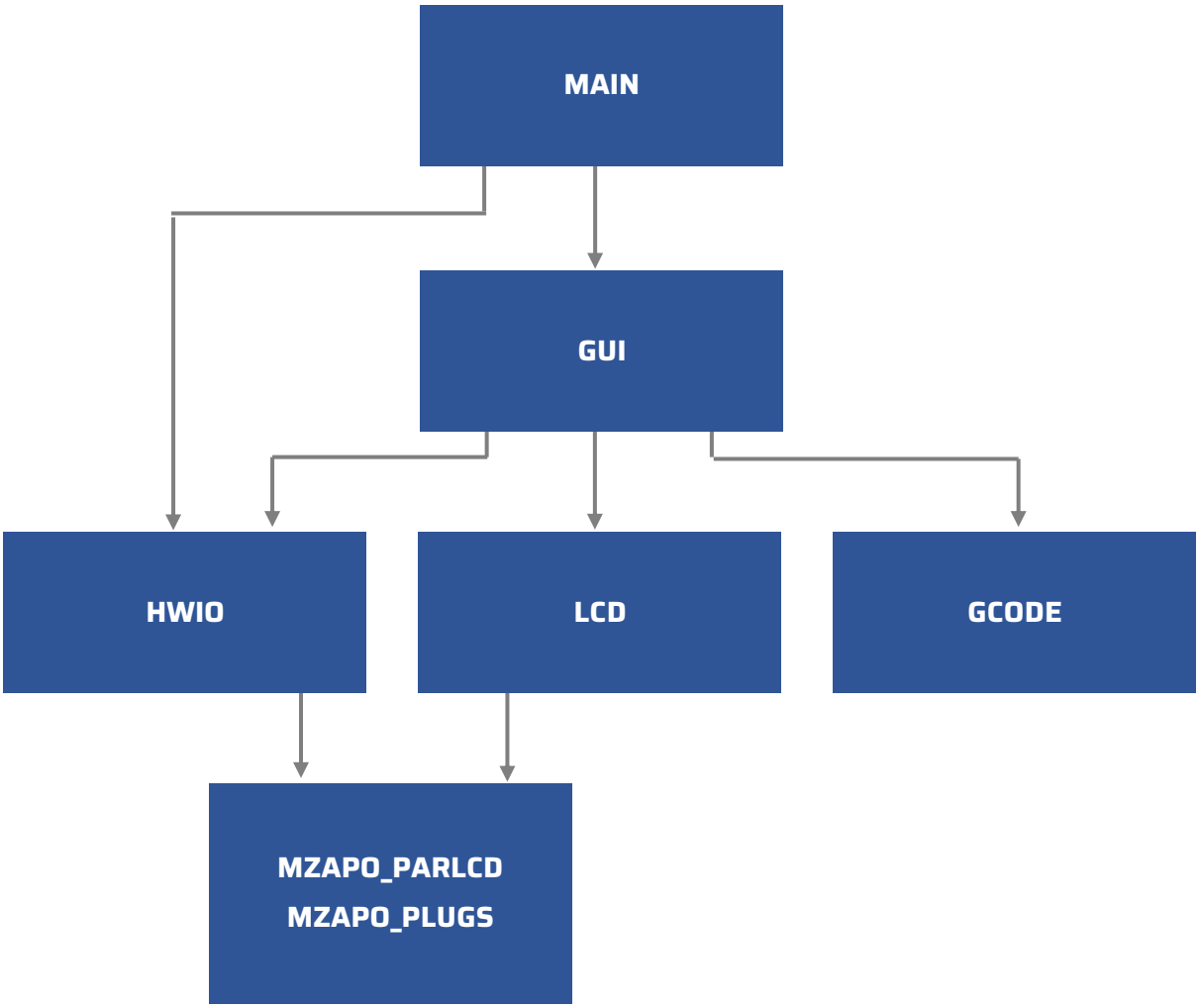
Extreme Gcode Analyzer

# Úvod

Extreme Gcode Analyzer je program běžící na přípravku MicroZed APO schopný otevřít a analyzovat gcode soubory pro 3D tiskárny. Kromě obecných informací o modelu dokáže na LCD displeji zobrazovat jednotlivé vrstvy. Zdrojový kód je dostupný v git repozitáři na github.com/humpljir/mzapo.

# Architektura aplikace

Blokové schéma jednotlivých modulů a naznačení komunikace mezi nimi.



# Funkce

Přiřazení funkcí podle jejich účelu ke konkrétním modulům programu.

## HWIO

```
bool hw_init(void);
void hw_check(void);
bool hw_r_released(void);
bool hw_g_released(void);
bool hw_b_released(void);
bool hw_r_incremented(void);
bool hw_g_incremented(void);
bool hw_b_incremented(void);
bool hw_r_decremented(void);
bool hw_g_decremented(void);
bool hw_b_decremented(void);
void hw_test_out(void);
void hw_write_ledstrip(uint32_t bits);
void hw_write_led1(uint8_t red, uint8_t green, uint8_t blue);
void hw_write_led2(uint8_t red, uint8_t green, uint8_t blue);
```

## MAIN

```
int main(int argc, char **argv);0
```

## LCD

```
void gcode_init_cmd(command *cmd);
bool gcode_parse_line (command *cmd, char *line, int size);
void gcode_process_cmd(command *cmd);
void gcode_print_cmd(command *cmd);
void gcode_move(command *cmd);
void gcode_print_stats(void);
bool gcode_init_file(char *filename);
void gcode_close_file(void);
void gcode_set_defualt_vals(void);
bool gcode_set_layer(layer_t *layer);
bool gcode_set_layer_by_num(int layer_num);
layer_t *gcode_get_layer(int layer_num);
void gcode_free_layer(layer_t *layer);
void gcode_print_layer(layer_t *layer);
bool gcode_get_cmd(command *cmd, bool move_only);
model_t *gcode_get_model(void);
file_t *gcode_get_file_info(void);
```

## LCD

```
bool lcd_init(void);
bool lcd_destroy(void);
void lcd_test(uint16_t color);
void lcd_paint(uint16_t color);
void lcd_write_pixel(disp_pos_t pixel, uint16_t color);
void lcd_paint_buffer(uint16_t color);
void lcd_draw_line(***);
void lcd_print_char(***);
void lcd_print_string(***);
unsigned char lcd_get_char_width(***);
int lcd_get_string_width(***);
void lcd_print_frame_buffer(void);
bool lcd_print_from_file(char *filename);
```

## GUI

```
void gui_win_right(void);
void gui_win_left(void);
void gui_win_manual(void);
void gui_win_manual_exit(void);
void gui_layer_up(void);
void gui_layer_down(void);
void gui_layer_up_4x(void);
void gui_layer_down_4x(void);
void gui_layer_up_16x(void);
void gui_layer_down_16x(void);
void gui_apply_state(void);
void gui_r_click(void);
void gui_r_incr(void);
void gui_r_decr(void);
void gui_g_click(void);
void gui_g_incr(void);
void gui_g_decr(void);
void gui_b_click(void);
void gui_b_incr(void);
void gui_b_decr(void);
void gui_set_quit(void);
void print_win1(void);
void print_win2(void);
void print_win3(void);
void print_win_man(void);
void gui_refresh_ledstrip(void);
bool gui_start(char *filename);
bool gui_init_file(char *filename);
bool gui_destroy(void);
void gui_print_layer(void);
disp_pos_t gui_map_extruder_to_disp(pos_t pos);
bool gui_quit(void);
```